CETE Méditerranée DCEDI/DERIS/TIER

MobiGIS

Démonstrateur libre

Analyse géographique de réseaux de voirie et Transports Collectifs

Spécifications fonctionnelles et techniques

version du 19/10/09

Mises à jour du rapport

Numéro de Version	Date	Auteur principal	Résumé des modifications
1	26/11/2008	L. Dezou (MobiGIS)	Création
2	30/12/2008	L. Dezou (MobiGIS)	Intégration des remarques de P. Gendre (CETE)
3	27/07/2009	C. Bloudeau (MobiGIS)	Evolution de la projection en Lambert II et prise en compte de la voirie OpenStreetMap
4	19/10/09	P. Gendre (CETE Med)	fusion docs installation, introduction

Contacts

Organisme/Société	Nom du correspondant/Adresse/Téléphone/E-mail
CETE Méditerranée	Nom du correspondant : Patrick Gendre Adresse: Avenue Albert Einstein CS 70499, 13593 Aix-en-Provence Cedex 3 Téléphone : 04 42 24 76 87 E-mail : pat.gendre@developpement-durable.gouv.fr
MobiGIS	Nom du correspondant : Laurent Dezou Adresse: Rue du Lanoux 31330 Grenade Téléphone : 05 81 60 80 82 E-mail : ldezou@mobigis.fr

Sommaire

I. INTRODUCTION 4
A. REQUÊTES SPATIALES ET SIG TRANSPORT OPEN SOURCE : LE PROJET POTIMART 4
B. LES DONNÉES 4
1. LES DONNÉES DE VOIRIE 5
2. LES DONNÉES DE TRANSPORT COLLECTIF 5
C. LES LOGICIELS 7
1. BASE DE DONNÉES: POSTGRESQL / POSTGIS 7
2. SYSTÈME D'INFORMATION GÉOGRAPHIQUE QGIS 7
3. AUTRES LOGICIELS 7
4. LES REQUÊTES VOIRIE / TC 8
D. PERSPECTIVES 9
E. PRODUCTION DE CARTES THÉMATIQUES ET D'ANALYSES 10

II. INSTALLATION DU DÉMONSTRATEUR 10
A. PYTHON 10
B. POSTGRESQL 11
C. POSTGIS 12
D. DÉCLARATION DES FONCTIONS PLPGSQL DANS LA BASE 16

III. CRÉATION ET INITIALISATION DE LA BASE DE DONNÉES 17
A. DONNEÉS ROUTIÈRES 17
1. TRANSFORMATION DES DONNÉES EN SHAPE FILES 18
2. EXPORT DES DONNÉES VERS LA BASE POSTGIS « CETE » 19
3. VÉRIFIER L'EXPORT DES DONNÉES VERS LA BASE « CETE » 20
4. COPIE DES CHEMINS DANS LA TABLE DES ROUTES 21
5. AJOUT DE LA COLONNE IMPASSE À LA TABLE ROUTE_AUTLSED31 21
B. DONNÉES CHOUETTE 21

IV. REQUÊTES DANS LA BASE POSTGRESQL / POSTGIS 22
A. REQUÊTES DANS LA BASE DE DONNÉES TC 22
1. AJOUT DE LA STRUCTURE GÉOMÉTRIE AUX TABLES CHOUETTE 22
2. MISE À JOUR DES GÉOMÉTRIES 22
3. CALCUL DES INDICATEURS 23
B. REQUÊTES DANS LA BASE DE DONNÉES VOIRIE 27
1. COPIE DES DONNÉES CHEMIN DANS LA TABLE DES ROUTES 27
2. RECHERCHE DES MAILLES 27
3. RECHERCHE DES IMPASSES 28
4. CALCUL DES INDICATEURS VOIRIE 29

V. DOCUMENT APPLICABLE ET RÉFÉRENCE 32

VI. GLOSSAIRE 32

Avant Propos

Remerciements

Nous tenons à remercier Alexandre Blaquière et toute son l'équipe (aujourd'hui intégrée à la Communauté Urbaine du Grand Toulouse) pour le soutien de TISSEO au projet POTIMART et en particulier pour nous avoir autorisés à utiliser un export des données d'offre TC dans le cadre de ce démonstrateur.

Résumé

Ce document est destiné aux techniciens des services Transport des collectivités, et plus généralement aux organismes et prestataires de service travaillant sur des études des réseaux de transport. Il présente un démonstrateur permettant de produire des analyses géographiques simples à partir d'une base de données décrivant des réseaux de voirie routière et de transport collectif. Ce démonstrateur est diffusé librement sur support DVD ou par Internet[1].

Le démonstrateur fonctionne avec des données réelles provenant du réseau de transport collectif Tisseo à Toulouse, partenaire du projet. La donnée de voirie provient des données Open Street Map. Utilisant des formats standard, il pourrait être facilement adapté à d'autres villes.

A terme, l'objectif de ce démonstrateur est de valider l'intérêt des utilisateurs potentiels (collectivités autorités de transport et autres organismes publics, bureaux d'études) pour une solution *open source* SIG d'analyse de réseaux de transport, et de mieux comprendre leurs besoins.

Ce démonstrateur ayant vocation à être complété et amélioré, nous invitons toute personne intéressée à nous faire part de ses remarques, questions, critiques ou suggestions[2].

Le développement de ce démonstrateur a été confié à la société MOBIGIS par le CETE Méditerranée fin 2008, dans le prolongement d'un projet de recherche[3] sur les SIG transport « open source ». La 1ère phase de cette étude a permis de développer des requêtes ou scripts SQL permettant de produire et des indicateurs puis des cartes relatifs d'une part à l'accessibilité piétonne des réseaux routiers (calculs d'impasses et de mailles), d'autre part à la fréquence et la vitesse des lignes d'un réseau TC. La 2ème phase a pour but de produire le démonstrateur, diffusable librement à partir de ces éléments.

La documentation associée à ce démonstrateur comprend 2 documents :

- le manuel d'utilisation permettant à un utilisateur final de faire fonctionner le démonstrateur ;
- les présentes spécifications fonctionnelles et techniques permettant à un utilisateur avancé, ou à un utilisateur souhaitant en savoir plus, de comprendre comment le démonstrateur a été conçu et de reproduire le processus de traitement des données aboutissant au démonstrateur, et éventuellement d'adapter ou compléter les requêtes pour les faire fonctionner sur d'autres jeux de données ou produire d'autres indicateurs et couches thématiques.

Les requêtes SQL mises au point permettent :

· de prendre en compte la composante spatiale des données TC au profil Chouette/Trident,
· d'intégrer les données voirie BD Topo dans la base PostgreSQL/PostGIS,
· de calculer des indicateurs prédéfinis.

A la suite de ces traitements, les résultats peuvent être cartographiés grâce à l'utilisation du logiciel SIG libre Quantum GIS (QGIS). Les requêtes sont testées avec des données réelles existantes à Toulouse.

Le document de spécifications :

présente une vision d'ensemble sur l'architecture du démonstrateur
· lister les constituants de la solution et décrit leur installation
· décrit les indicateurs et les calculs opérés
· fournit le mode d'utilisation des scripts
donne quelques perspectives d'amélioration du démonstrateur.

[1] sur les site http://www.cete-mediterranee.fr/tt13/, www.predim.org et www.potimart.org

[2] par e-mail en contactant pat.gendre@developpement-durable.gouv.fr et ldezou@mobigis.fr

[3] www.potimart.org

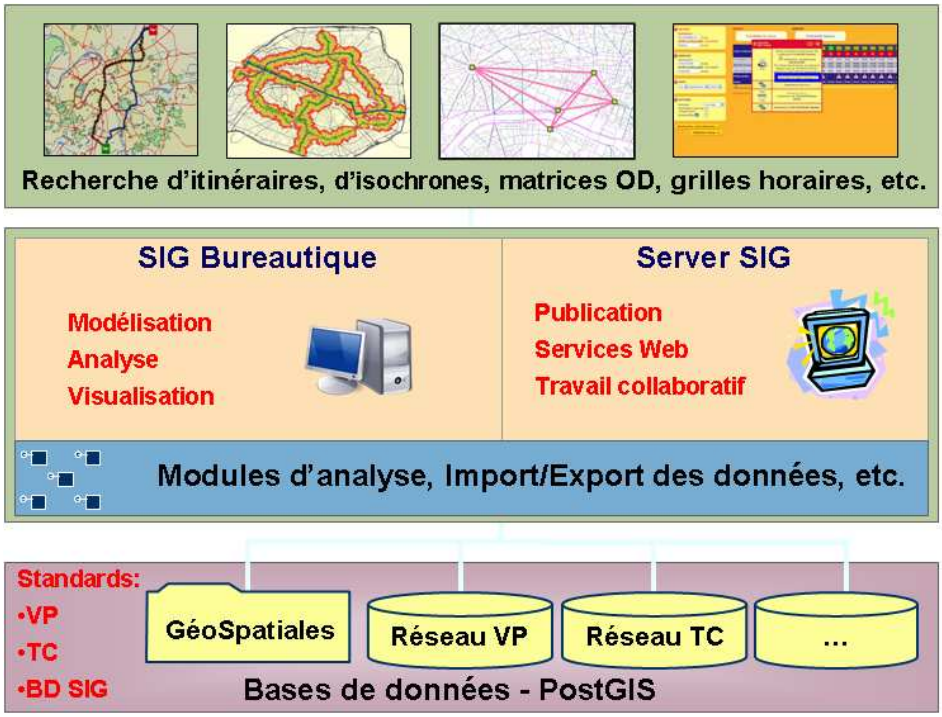
I. INTRODUCTION

A. Requêtes spatiales et SIG transport open source : le projet POTIMART

Le développement de ce démonstrateur a été confié à la société MOBIGIS par le CETE Méditerranée fin 2008, dans le prolongement d'un projet de recherche sur les SIG transport « open source ». La 1ère phase de cette étude a permis de développer des requêtes ou scripts SQL permettant de produire et des indicateurs puis des cartes relatifs d'une part à l'accessibilité piétonne des réseaux routiers (calculs d'impasses et de mailles), d'autre part à la fréquence et la vitesse des lignes d'un réseau TC. La 2ème phase a pour but de produire un démonstrateur diffusable librement à partir de ces éléments.

Ce démonstrateur s'inscrit dans la continuité du projet de recherche Potimart[4], financé dans le cadre du programme Predim[5], qui a pour objectif de développer des solutions open source (logiciels libres) pour l'analyse géographique de réseaux de transport. Les deux premières phases ont été réalisées en 2008. Les résultats du projet sont publiés sur le site www.potimart.org La phase 3, d'une durée de 6 mois au 2ème semestre 2009, a pour buts d'approfondir l'analyse des besoins, de consolider les composants logiciels, et d'identifier un site pilote.

Le schéma ci-dessous donne une vue d'ensemble de l'architecture dans laquelle s'inscrit le démonstrateur.



B. Les données

Le démonstrateur comprend 2 principaux types de données, celles concernant la voirie d'une part, d'autre part celles décrivant les réseaux de TC.

Dans une prochaine version, il serait intéressant de compléter le démonstrateur par des données de recensement de population Insee et les indicateurs que l'on peut en tirer.

[4] www.potimart.org

[5] www.predim.org

1. Les données de voirie

Open Street Map est une initiative privée visant à produire une cartographie vectorielle en libre-diffusion du réseau routier mondial. Partie d'Angleterre, OSM a désormais largement franchi les frontières et la qualité des données OpenStreetMap (OSM) s'est améliorée de manière spectaculaire à partir de 2008, comme en témoigne l'ouverture du service OpenRouteService, qui démontre la mise en oeuvre d'un service de calcul d'itinéraire routier s'appuyant sur les données OSM et les spécifications OGC OpenLS. Grâce à des outils de saisie très évolués et à l'effort des bénévoles, les données disponibles en France, notamment autour de Toulouse, sont désormais assez complètes.

La qualité des tronçons du réseau OSM est tout à fait correcte, d'autant plus que les bénévoles d'OSM ont obtenu l'accord du ministère des finances pour s'appuyer sur le fond de plan cadastral diffusé sur le web, ce qui permet une saisie du réseau en ligne, sans faire de relevé GPS sur le terrain.

Une difficulté en revanche sur les tronçons OSM est qu'ils ne sont pas forcément tracés d'une intersection à une autre mais peuvent représenter toute une rue, et croiser d'autres tronçons : ce défaut nous a demandé de faire un pré-traitement avant de pouvoir effectuer la détection des mailles et des impasses. Heureusement, un script SQL de création d'une couche de points d'intersections entre les tronçons d'une couche polyligne avait déjà été développé par la communauté postgis, et notre script l'utiliser, ce qui l'a largement simplifié. Le script fonctionne correctement (en tout cas sans produire d'erreur), mais il semble que certains cas particuliers de géométries n'ont pas été traités comme il l'aurait fallu. Nous n'avons pas eu le temps d'approfondir ce point, en gardant à l'esprit que ce démonstrateur vise à montrer les principes des outils et n'ambitionne une qualité parfaite de données.

A noter que la démonstration pourrait fonctionner avec d'autres types de données, en particulier avec la BD Topo de l'IGN, ce qui présenterait plusieurs intérêts : sa précision, sa couverture nationale, le fait que beaucoup de collectivités et de services de l'état (y compris le ministère du développement durable et le CETE) ont la licence d'utilisation.

2. Les données de transport collectif

Dans le cadre de groupes de travail nationaux et européens de normalisation de l'information transport, un modèle de données ainsi que des profils d'échange[6] de données XML ont été spécifiés pour les données décrivant l'offre théorique d'un réseau de transport collectif. Le ministère a financé dans ce cadre le développement d'un logiciel *open source* implémentant ce profil d'échange, l'application Chouette, qui permet donc des imports / exports et la gestion d'une Base de Données d'offre théorique TC.

L'objectif de ce démonstrateur de requêtes Voirie/TC est aussi de valoriser ces formats d'échange standard : dans la mesure où les données sont potentiellement disponibles à terme sur l'ensemble des réseaux français, les mêmes bases de données, requêtes et outils d'analyse devraient pouvoir en effet être mis en oeuvre sans difficulté.

En l'occurrence, les données utilisées proviennent du réseau TC toulousain Tisseo et ont été fournies par l'autorité organisatrice de transport urbain (Communauté Urbaine du Grand Toulouse). Elles nous été livrées directement en tant que tables postgreSQL (sous forme de fichier « backup »). Elles sont issues d'un export de fichiers XML au profil Trident/Chouette produit par l'exploitant du réseau TC et réimportées dans l'outil Chouette (qui utilise la base postgreSQL/postgis).

L'application Chouette est disponible sur le site www.chouette.mobi ; un marché de maintenance, d'évolution et d'accompagnement a été lancé par le CERTU en septembre 2009, l'application devrait donc bénéficier d'amélioration dès début 2010. Il faut savoir que la description complète des horaires d'un réseau de TC est assez complexe ; elle s'appuie sur un modèle conceptuel de données normalisé au niveau européen baptisé Transmodel, qui comprend des notions génériques qui décrivent un réseau de transports en commun, et que l'on retrouve sous forme de tables dans la base CHOUETTE, par

[6] Le profil baptisé Trident/Chouette va désormais s'appeler Neptune dans le cadre son évolution soumise fin 2009 au vote du groupe AFNOR/BNEVT CN03/GT7.

exemple :

- · ***Réseau (PTNetwork)***

Le réseau représente les lignes de transport qui le constituent. Cette notion renseigne sur l'identification du réseau.

- · ***Transporteur (Company)***

Il s'agit de l'exploitant de la ligne. Chaque transporteur dispose d'un identifiant unique pour toute la base CHOUETTE, quel que soit le réseau ; il est donc important que le gestionnaire de la base CHOUETTE le renseigne d'une manière bien définie et pérenne.

- · ***Ligne (Line)***

Une ligne se définit par des propriétés telles que l'indice et le nom, qui sont connues des usagers. Une ligne référence un réseau et un transporteur ; par ailleurs la ligne se compose d'un ou plusieurs itinéraires.

- · ***Itinéraire (ROUTE)***

L'itinéraire est une sélection ordonnée de références aux points d'arrêt. Un itinéraire est spécifique à une ligne, mais les itinéraires peuvent référencer des arrêts communs.

- · ***Course (VehiculeJourney) et Horaire (VehicleJourneyAtStop)***

Une course décrit le déplacement d'un véhicule de transport public sur un itinéraire de la ligne. La course parcourt les arrêts de l'itinéraire dans l'ordre, sans nécessairement s'arrêter à chacun des arrêts. La course est rattachée à un nombre variable de calendriers d'application.

- · ***Mission (Journey Pattern)***

Si on considère la suite ordonnée des arrêts d'un itinéraire, la mission se définit comme une suite. Autrement dit, toutes les courses d'un itinéraire qui desservent les mêmes arrêts et dans le même ordre, à des horaires différents éventuellement, référencent la même mission.

- · ***Correspondance (ConnectionLink)***

La correspondance permet d'établir des liaisons entre les arrêts. Les liaisons définissent une durée de parcours entre les arrêts ou zones reliées. La description des données permet d'appréhender la composition des tables de la base de données.

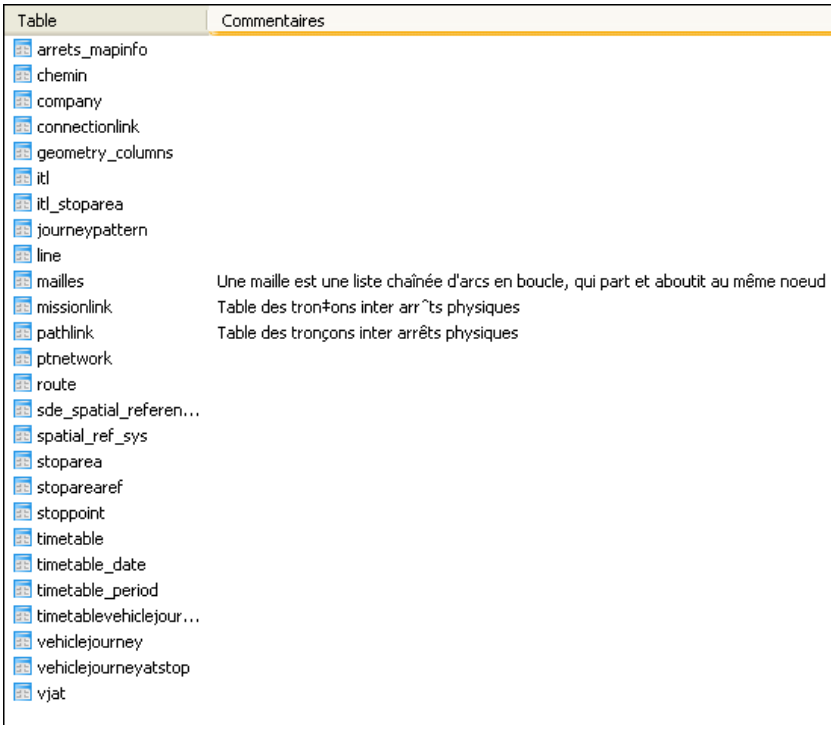


Table	Commentaires
arrets_mapinfo	
chemin	
company	
connectionlink	
geometry_columns	
itl	
itl_stoparea	
journeypattern	
line	
mailles	Une maille est une liste chaînée d'arcs en boucle, qui part et aboutit au même noeud
missionlink	Table des tron‡ons inter arr^ts physiques
pathlink	Table des tronçons inter arrêts physiques
ptnetwork	
route	
sde_spatial_referen...	
spatial_ref_sys	
stoparea	
stoparearef	
stoppoint	
timetable	
timetable_date	
timetable_period	
timetablevehiclejour...	
vehiclejourney	
vehiclejourneyatstop	
vjat	

copie d'écran listant les tables d'une base Chouette

En pratique, les tables que nous avons directement utilisées, car représentables sous forme de

couches SIG, sont surtout les tables **Stoparea** (points d'arrêt) et **MissionLink** (tronçons de ligne).

Les tables *timetable* contiennent les calendriers d'application des horaires de transports. Ils permettent de définir les horaires d'une course.

Toutes ces données doivent être renseignées par le gestionnaire du réseau, qui utilisé en général pour cela des outils informatiques, dits de « production horaire ».

C. Les logiciels

Les traitements ont été effectué pour l'essentiel avec des logiciels open source : base de données et client SIG.

1. Base de données: PostgreSQL / postgis

La base de données PostgreSQL est la principale base de données relationnelles *open source* avec MySQL et SQLlite. PostgreSQL dispose d'une extension géographique Postgis, qui a pratiquement le monopole dans le domaine *open source*.

Cette extension permet de décrire la géométrie des objets gérés dans les tables, de manière à pouvoir les visualiser sous forme de «*couche*» dans un Système d'Information Géographique (SIG), et à pouvoir faire des requêtes spatiales et topologiques (proximité, inclusion, etc.)

Les performances de Postgresql permettent de traiter de gros volumes. PostgreSQL se prend rapidement en main pour les fonctionnalités de base, grâce à l'outil graphique PgAdmin. L'aide pour la réalisation des requêtes est accessible (mais en anglais). Le langage SQL permet facilement d'effectuer des requêtes simples de manière interactive. Postgis comprend un certain nombre de fonctions permettant en outre d'effectuer des requêtes spatiales sur les données (proximité, intersection, etc.).

En revanche, dès qu'on veut obtenir des réponses à des requêtes plus complexes, la syntaxe de SQL est difficile et son débuggage problématique, a fortiori quand on a besoin d'écrire des fonctions (en langage PL/SQL ou python). L'utilisation d'un langage de haut niveau devient alors plus appropriée.

2. Système d'information géographique QGIS

Quantum GIS (QGIS) est un des clients SIG *open source* les plus répandus, qui permet très facilement de se connecter à une base Postgis, un point essentiel dans notre cas. QGIS gère de nombreux formats de fichiers. Il peut travailler des images vectorielles et raster, ainsi que des couches postgis. L'existence de formats standard pour les données géographiques (shapefile, etc.) fait que d'autres outils commerciaux auraient pu aussi être utilisés (ArcMap, Mapinfo, etc.) pour visualiser les données de la base postgis, et permettraient éventuellement des fonctionnalités plus élaborées de cartographie dont nous n'avions pas besoin dans le cadre de ce stage.

Les principaux atouts de ce logiciel sont qu'il est simple à l'utilisation et donc abordable par une personne ayant une connaissance minimum des SIG. QGIS intègre un outil de mise en page.

3. Autres logiciels

a) Machine virtuelle Virtual Box

Virtual Box est un logiciel permettant d'installer une machine virtuelle sous Windows ou Linux. Il est utilisé afin de livrer un démonstrateur pré-installé, avec les bonnes versions et le bon paramétrage des logiciels, les requêtes SQL déjà installées, les données déjà importées en base.

La démonstration tourne sous linux (ubuntu) puisqu'elle doit pouvoir être diffusée librement.

b) CHOUETTE

Ce logiciel *open source* est utilisé en amont pour produire les données TC de manière standardisée (cf. plus haut). L'outil Chouette (***C**réation d'**h**oraires avec **u**n **OU**til d'**É**change de données **TC** selon le format **T**rident **E**uropéen*) a été réalisé par la société Dryade pour le compte du CERTU sur financement PREDIM.

Actuellement, l'outil se limite aux échanges de données concernant la description statique de l'offre de transports en commun et ne couvre pas les messages de perturbation, par exemple.

Les objectifs affichés de cet outil sont de:

- Donner aux utilisateurs qui le souhaitent la possibilité de tester les profils d'échange XML/TRIDENT[7] sur des données réelles et de fournir des propositions d'amélioration pouvant être prises en compte dans le cadre de la normalisation.
- Transformer les fiches horaires de petits exploitants aux formats TRIDENT, et générer un mini site web par la même occasion si cela s'avère une nécessité pour l'exploitant.
- Permettre à de petites autorités organisatrices de transport de gérer des centrales d'information
- Donner aux exploitants de réseaux et autres acteurs la possibilité de télécharger le logiciel lui-même pour pouvoir l'analyser et l'intégrer à leurs systèmes d'information.

c) autres outils

Nous avons utilisé notamment pour les conversions de systèmes de projection et les imports/exports l'outil ogr2ogr, qui est un logiciel open source de traitement de données (d'ailleurs utilisé aussi dans QGIS).

Le logiciel statistique open source R (www.r-project.org) peut être un outil intéressant pour effectuer certains calculs plus facilement qu'en SQL. Il se connecte facilement à PostgreSQL.

4. Les requêtes Voirie / TC

Les requêtes seront décrites plus en détail dans la partie IV.

a) Requêtes voirie : mailles et impasse

L'objectif de ces requêtes est de calculer des indicateurs qui renseignent sur la *perméabilité* (la facilité à traverser le réseau de voirie). Elles permettent de qualifier la facilité de se déplacer à pied (ou à vélo) dans un quartier, dans la mesure où ces modes de déplacement dits « doux » (ou « actifs ») sont très sensibles aux détours, aux « effets de coupure » qui empêchent de prendre la ligne droite pour aller d'un point à un autre. Le taux d'impasses, ou le périmètre des pâtés de maison (ilots, blocs, ou mailles, selon le vocabulaire) sont les indicateurs calculés. Il est clair qu'un quartier dont le périmètre moyen des mailles est par exemple 700 mètres (10 minutes à pied) ne sont guère pratiqués par les piétons.

Cet indicateur mesure donc aussi indirectement la facilité d'accès aux TC, puisqu'en général les usagers vont à pied prendre le bus.

Le réseau piéton n'est pas forcément confondu avec le réseau routier accessible aux voitures. A minima nous avons supprimé pour la démonstration le réseau autoroutier (interdit aux piétons) des données OSM. Une analyse plus fine des cheminements impliquerait un travail très lourd de vérification sur le terrain. Pour cette démonstration, l'important est de démontrer la faisabilité des outils, l'objectif n'est pas une étude opérationnelle sur la voirie urbaine.

En pratique, l'utilisation des requêtes est le suivant.

Dans un premier temps, une table MAILLES est créée, contenant un polygone par maille (îlot). La requête de création des mailles s'appuie sur la fonction postgis *polygonize* pour créer à partir d'une couche

[7] Le profil Trident/Chouette devrait être prochainement (début 2010) rebaptisé ... Neptune.

de lignes (la table ROUTE contenant la voirie étudiée) les mailles du réseau de voirie. Il est ensuite assez simple d'en calculer le périmètre ou la surface.

Dans un second temps, un script SQL ajoute une colonne « impasse » à la table de voirie (ROUTE), colonne pouvant prendre la valeur true/false. La fonction PL/SQL isdeadend évalue l'appartenance de chaque tronçon de voirie à une impasse.

A partir de ces éléments, on peut calculer des indicateurs sur les mailles et impasses, sur un territoire donné (par exemple sur une commune, ou sur les zones IRIS de l'Insee) :

- taux d'impasses en %

- valeur min, max, moyenne, médiane du périmètre des mailles

Il est ensuite possible de visualiser ces indicateurs sur un SIG, en créant des cartes thématiques.

b) Requêtes TC : fréquences et vitesse moyenne

L'objectif des fonctions de requêtes développées par Mobigis est de produire des indicateurs de base sur tous les arrêts du réseau TC, pour une tranche horaire et un jour donnés :

- vitesse moyenne d'une ligne, tronçon par tronçon ;
- fréquence de passage à un arrêt pour une ou plusieurs lignes.

Le calcul du nombre de bus passant à un arrêt pendant une période (jour, tranche horaire) donnée consiste à compter le nombre de courses (*vehicle journey*). Ce nombre est inséré comme une colonne attributaire JOURNEYNB supplémentaire dans la table STOPAREA.

Le calcul des vitesses moyennes est compliqué par le fait que tous les bus ne passent pas à tous les arrêts (il y a des lignes, directes, semi-directes, express, etc.) et la notion de tronçons n'est donc pas immédiatement disponible en base : il faut d'abord créer la table des tronçons (MissionLink). Par ailleurs, les horaires étant arrondis à la minute près, ils ne sont pas significatifs entre 2 arrêts consécutifs. Les vitesses moyennes sont donc calculées comme des moyennes glissantes sur plusieurs tronçons. Cette fonction plus complexe a été écrite en python.

D. Perspectives

Lors d'un stage de Master effectué au CETE d'avril à juin 2009 par Claire Christol, nous avons cherché à compléter les données en s'appuyant sur celles diffusées par l'Insee sur son site Internet, issues du recensement de la population. Nous n'avons utilisé que les population et nombre de logements dans les données Insee, mais il serait possible d'utiliser des données plus fine (par sexe, par classe d'âge, éventuellement par catégorie socio-professionnelle) pour affiner les indicateurs et les analyses.

Nous avons complété les scripts SQL en vue de pouvoir réinstaller complètement la base VP/TC, créer les tables mailles, impasses, arrêts, tronçons, ainsi que la table IRIS des zones Insee.

Cette table comporte une ligne par zone IRIS ; nous avons ajouté des colonnes supplémentaires en vue de produire des cartes thématiques, sur des attributs dérivés des calculs d'indicateurs sur la voirie et les TC :

- pourcentage d'impasses (c'est-à-dire longueur totale des impasses divisé par longueur totale de voirie) par IRIS

- périmètre moyen des mailles comprises dans un IRIS

- nombre d'arrêt, nombre de passages de bus, vitesse moyenne par IRIS, densité d'arrêts ou de passage à l'hectare..

Les améliorations proposées concernent donc notamment l'utilisation des données de population

diffusées par l'Insee, qui pourraient enrichir beaucoup l'intérêt du démonstrateur, en permettant leur croisement avec les données transport. Il serait nécessaire pour cela d'obtenir l'autorisation de diffuser les géométries (contours) des zones IRIS utilisées par l'Insee dans les recensements.

Les prochaines étapes envisagées par le CETE et ses partenaires Mobigis et Dryade dans le cadre du projet Potimart consistent à compléter le démonstrateur avec d'autres scripts, notamment d'utiliser la librairie postgreSQL pgrouting, qui permet des calculs de plus court chemin, améliorer les indicateurs et les représentations cartographiques, ainsi que de pouvoir accéder aux données de la base à partir d'un langage de haut niveau plus convivial que SQL, ou d'associer un serveur web à la base postgis. Ces développements ne nécessitent pas à proprement le développement de nouveaux logiciels, mais demandent néanmoins des adaptations, des tests et un effort assez lourd de mise au point, qui ne pourra être effectué qu'ultérieurement compte tenu des ressources très limitées affectées au financement de ce projet.

E. Production de cartes thématiques et d'analyses

Les tables produites ou complétées dans postgis sont ensuite accessibles depuis un SIG en tant que couches géographiques pour autant qu'elles possèdent une colonne de type « géométrie » (geometry) correctement renseignée. Une fois que l'on s'est connecté sur la base de données contenant les tables sur lesquelles on veut travailler.

Pour visualiser les données dans QGIS, il y a deux méthodes principalement :

- soit ouvrir un fichier Shapefile. Il peut être plus pratique selon les cas d'exporter les données en base sous cette forme. Il suffit d'utiliser le programme utilitaire ogr2ogr pour réaliser cet export en ligne de commande DOS.

- soit ouvrir directement une table en base de données : QGIS sait se connecter facilement à la base de données (à condition bien sûr que le serveur soit lancé, et avec le bon mot de passe !), et il détecte automatiquement les tables de la base qui possèdent une colonne géométrie correcte, qui permet de les ouvrir en tant que couche dans Postgis

On peut ensuite travailler à la fois avec des couches Shapefile ou Postgis, ou créer de nouvelles couches si besoin. Il ne reste plus qu'à composer des cartes suivant les analyses que l'on veut faire. Les différentes couches se superposent correctement, pour autant que le système de référencement spatial soit cohérent.

II. INSTALLATION DU DÉMONSTRATEUR

Liste des constituants et version utilisées :

Python	http://www.python.org	2.5.2
PostGreSQL	http://www.postgresql.org/	8.4.4
PostGIS	http://postgis.refractions.net	1.3.3
QuantumGIS ou QGIS	http://www.qgis.org	0.11
SPIT	Outil d'importation de shapefile vers PostGIS	Extension de QGIS

Le démonstrateur peut être installé sous Windows (sans machine virtuelle), comme expliqué ci-dessous. Il peut de manière très semblable être sous linux, y compris sur une machine virtuelle (logiciel Virtual Box (dont l'installation est décrite dans le manuel utilisateur, car nécessaire au fonctionnement du démonstrateur).

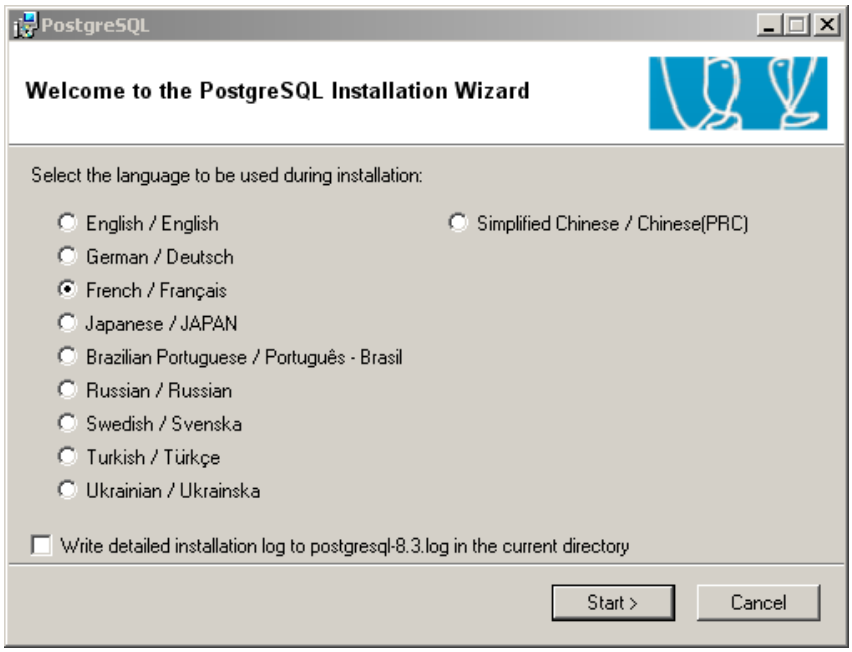
A. Python

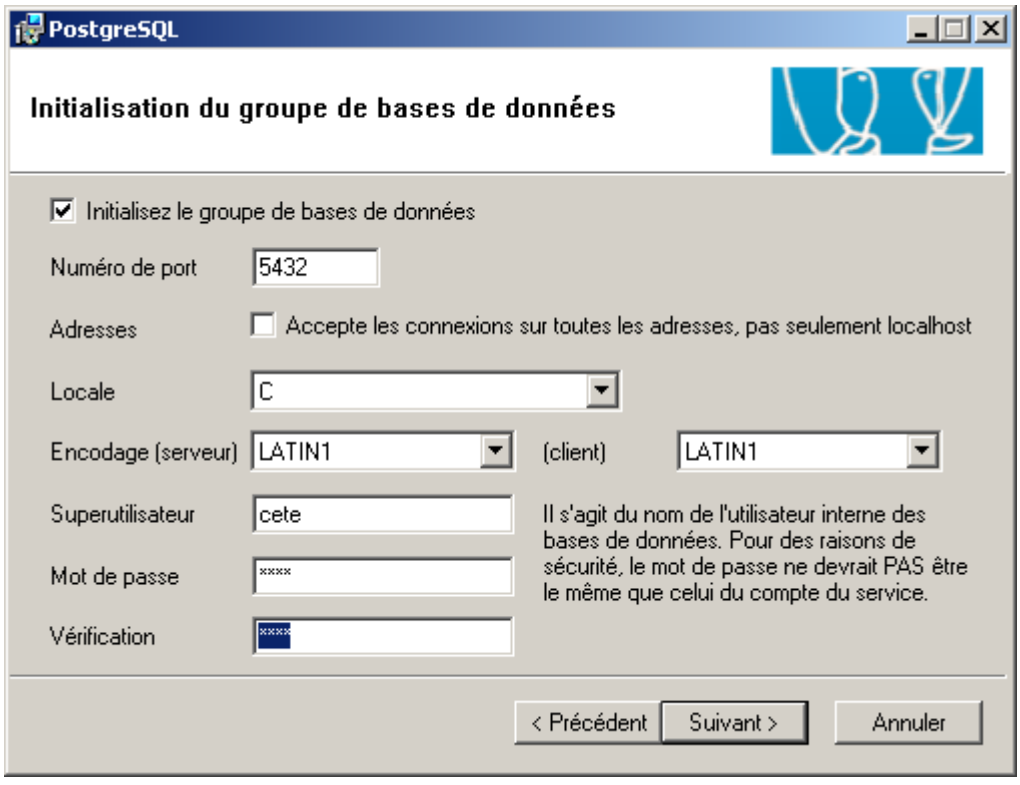
Pour utiliser le langage Pl/Python, il faut installer le langage sous windows à partir de python-2.5.2.msi

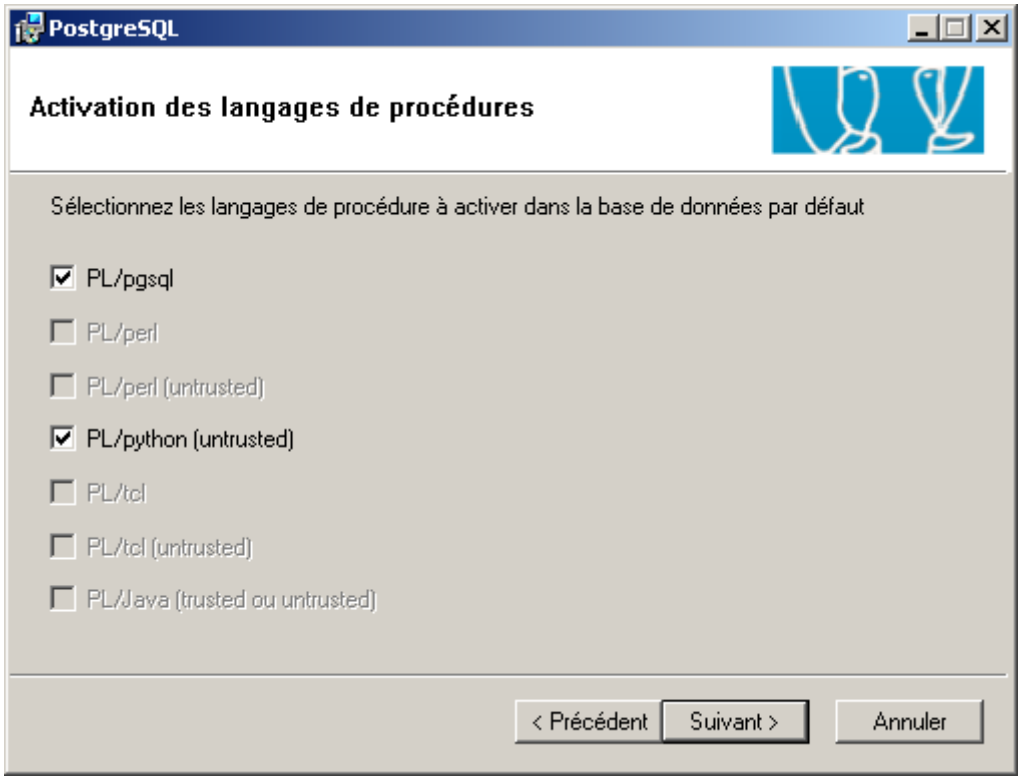
B. PostGreSQL

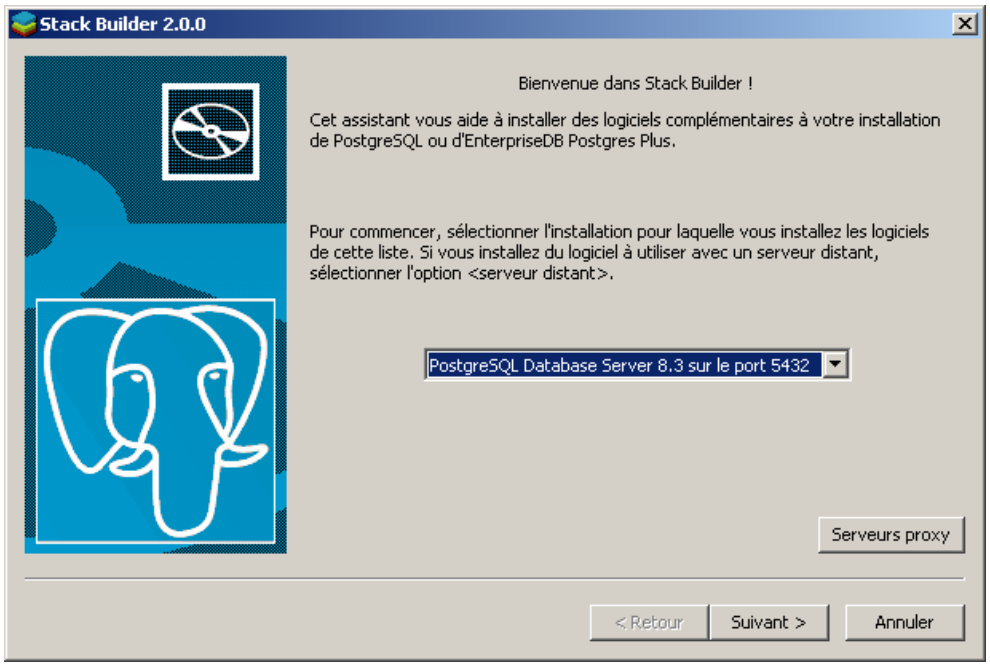
Manuel utilisateur : http://www.postgresql.org/docs/8.3/static/index.html

Installation à partir de postgresql-8.3.msi



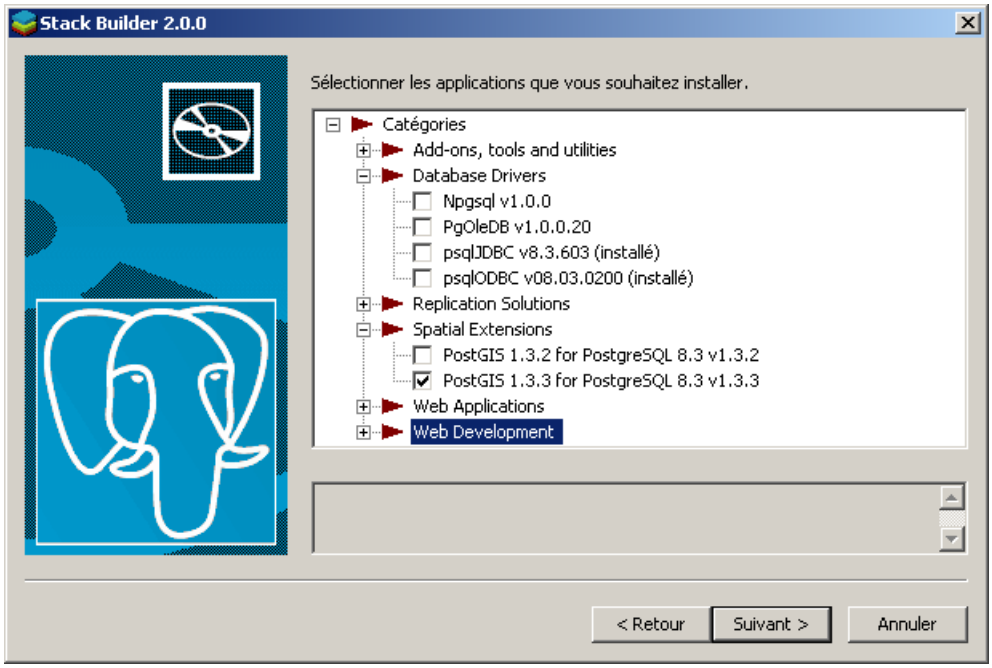

PostgreSQL
Initialisation du groupe de bases de données
Initialisez le groupe de bases de données
Numéro de port
5432
Adresses
Accepte les connexions sur toutes les adresses, pas seulement localhost
Locale
C
Encodage (serveur)
LATIN1
(client)
LATIN1
Superutilisateur
cete
Il s'agit du nom de l'utilisateur interne des bases de données. Pour des raisons de sécurité, le mot de passe ne devrait PAS être le même que celui du compte du service.
Mot de passe
Vérification
< Précédent
Suivant >
Annuler




Stack Builder 2.0.0
Bienvenue dans Stack Builder !
Cet assistant vous aide à installer des logiciels complémentaires à votre installation de PostgreSQL ou d'EnterpriseDB Postgres Plus.
Pour commencer, sélectionner l'installation pour laquelle vous installez les logiciels de cette liste. Si vous installez du logiciel à utiliser avec un serveur distant, sélectionner l'option <serveur distant>.
PostgreSQL Database Server 8.3 sur le port 5432
Serveurs proxy
< Retour
Suivant >
Annuler

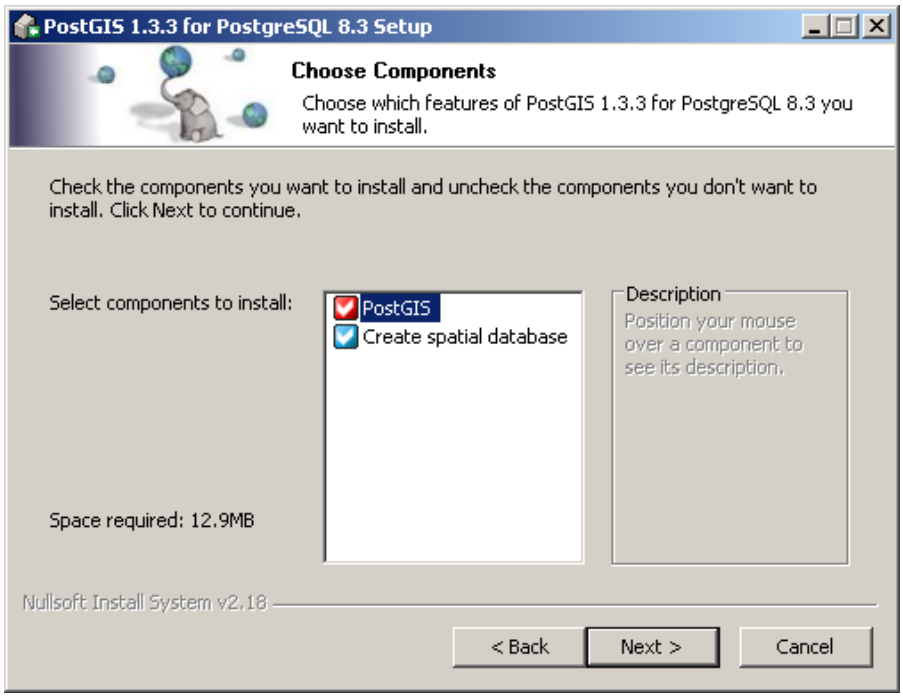
C. PostGIS

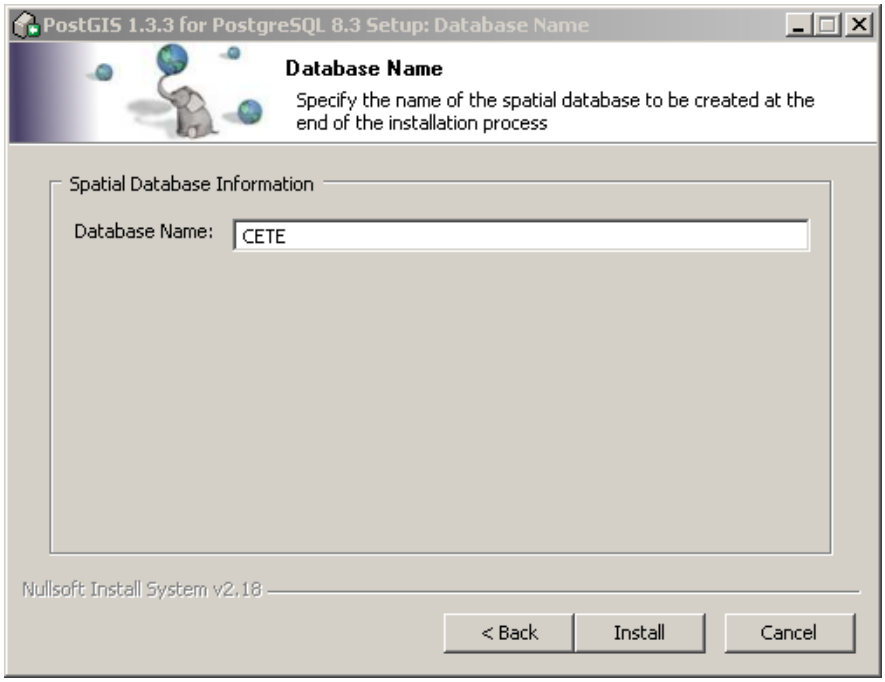
Installation à l'aide du Stack Builder PostgreSQL (Constructeur de la pile applicative) :



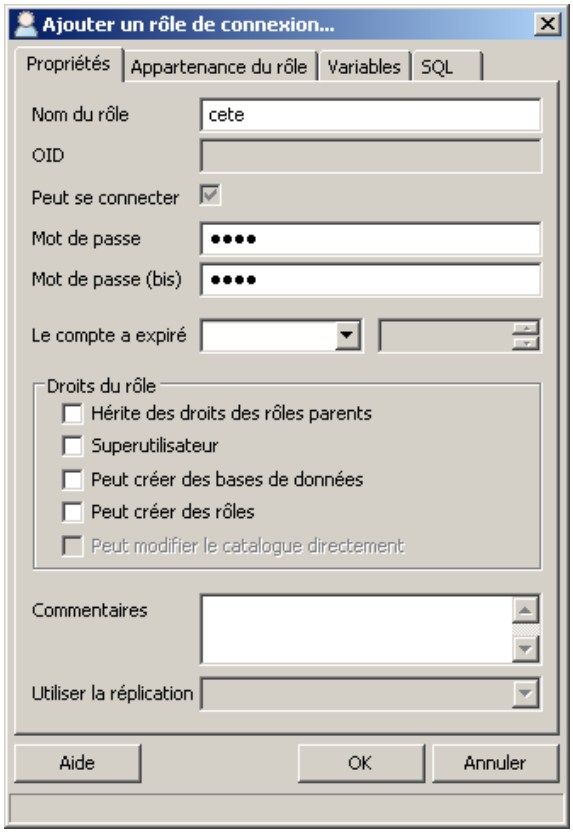
Création de la base PostGIS, deux possibilités :

- A la suite de l'installation PostGIS

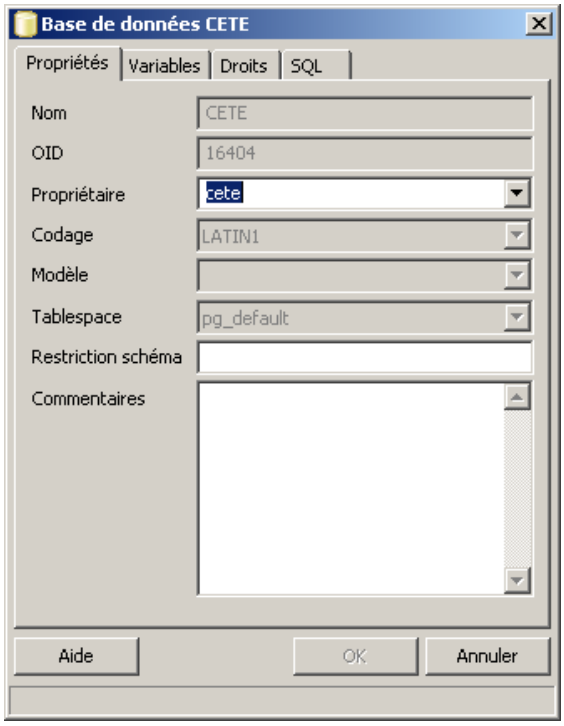

PostGIS 1.3.3 for PostgreSQL 8.3 Setup
Choose Components
Choose which features of PostGIS 1.3.3 for PostgreSQL 8.3 you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
PostGIS
Create spatial database
Description
Position your mouse over a component to see its description.
Space required: 12.9MB
Nullsoft Install System v2.18
< Back
Next >
Cancel


PostGIS 1.3.3 for PostgreSQL 8.3 Setup: Database Name
Database Name
Specify the name of the spatial database to be created at the end of the installation process
Spatial Database Information
Database Name:
CETE
Nullsoft Install System v2,18
< Back
Install
Cancel

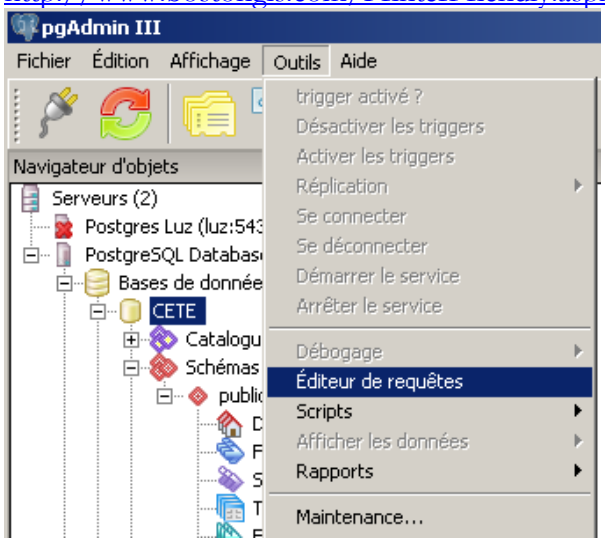
- En dehors de l'installation PostGIS :
  - Via PgAdmin ; création de l'utilisateur cete :



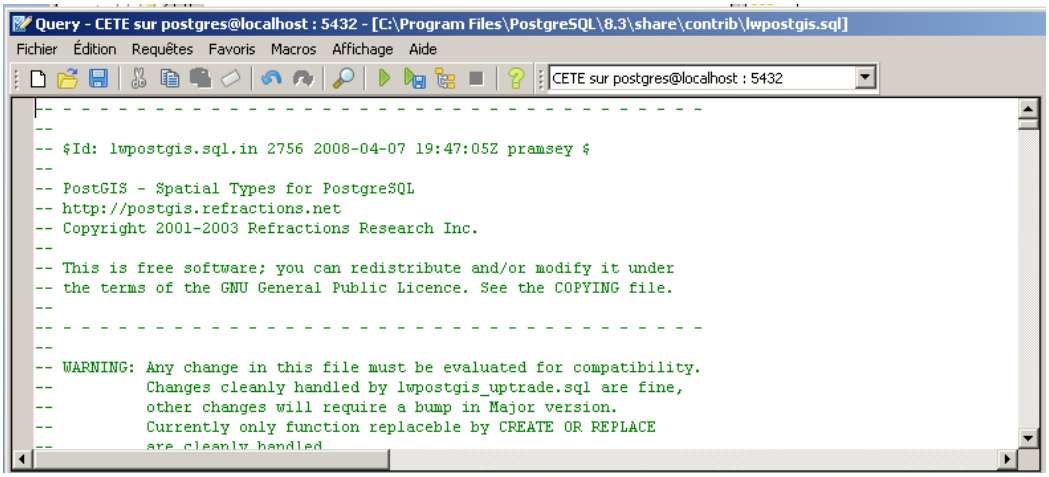
- Création de la base de données :



- Ajouter la composante spatiale à la base de données (fait conformément à http://www.bostongis.com/PrinterFriendly.aspx?content_name=postgis_tut01)



- Dans la fenêtre d'édition des requêtes, créer le language plpgsql : « **create language plpgsql** »
- Cliquer la flèche verte
- Ouvrir le fichier C:\Program Files\PostgreSQL\8.3\share\contrib\lwpostgis.sql


Query - CETE sur postgres@localhost : 5432 - [C:\Program Files\PostgreSQL\8.3\share\contrib\lwpostgis.sql]
Fichier Édition Requêtes Favoris Macros Affichage Aide
CETE sur postgres@localhost : 5432
-- $Id: lwpostgis.sql.in 2756 2008-04-07 19:47:05Z pramsey $
-- PostGIS - Spatial Types for PostgreSQL
-- http://postgis.refractions.net
-- Copyright 2001-2003 Refractions Research Inc.
-- This is free software; you can redistribute and/or modify it under
-- the terms of the GNU General Public Licence. See the COPYING file.
-- WARNING: Any change in this file must be evaluated for compatibility.
-- Changes cleanly handled by lwpostgis_uptrade.sql are fine,
-- other changes will require a bump in Major version.
-- Currently only function replaceble by CREATE OR REPLACE

- Cliquer la flèche verte
- Ouvrir le fichier C:\Program Files\PostgreSQL\8.3\share\contrib\spatial_ref_sys.sql
  A la fin de fichier sql, supprimer la ligne VACUUM ANALYZE spatial_ref_sys;
  Cliquer la flèche verte

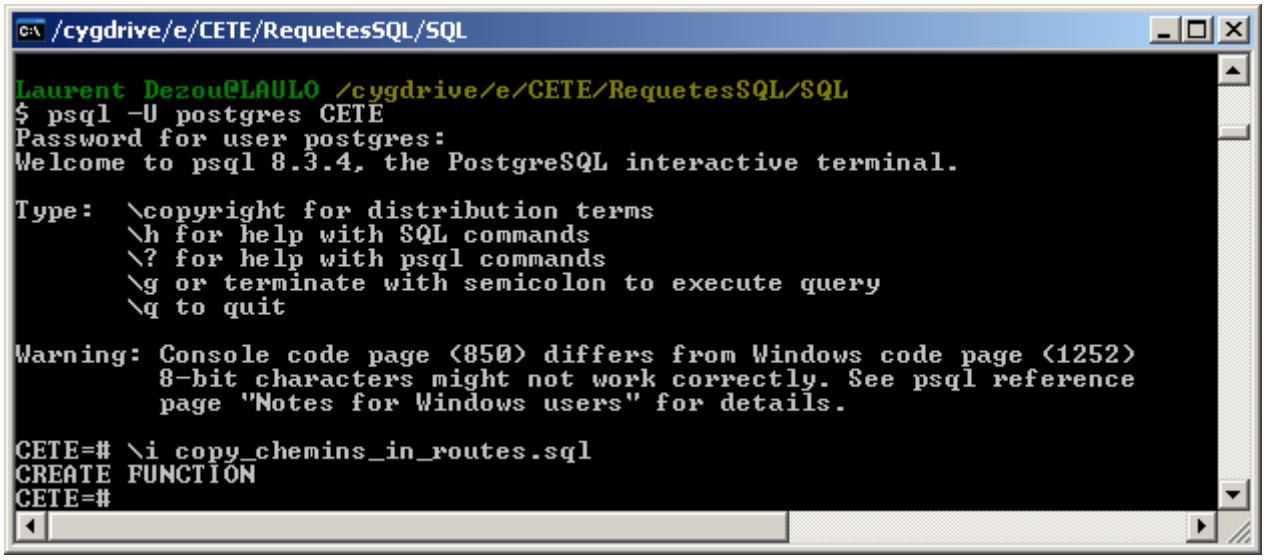
Pour permettre l'utilisation de Spit plugin et pour autoriser l'utilisateur « cete » de créer la géométrie des tables stopearea : GRANT ALL ON geometry_columns TO PUBLIC;

D. Déclaration des fonctions plpgsql dans la base

Les fonctions plpgsql sont disponibles dans le répertoire de livraison RequetesSQL.

Elles sont écrites pour l'utilisateur cete ; Dans le cas où la base est utilisée par un autre utilisateur, éditer les scripts et corriger le nom de l'utilisateur cete en le nom souhaité.

Dans une fenêtre de commande, se placer dans le répertoire de livraison RequetesSQL :


/cygdrive/e/CETE/RequetesSQL/SQL
Laurent Dezou@LAULO /cygdrive/e/CETE/RequetesSQL/SQL
$ psql -U postgres CETE
Password for user postgres:
Welcome to psql 8.3.4, the PostgreSQL interactive terminal.
Type: \copyright for distribution terms
\h for help with SQL commands
\? for help with psql commands
\g or terminate with semicolon to execute query
\q to quit
Warning: Console code page (850) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
CETE=# \i copy_chemins_in_routes.sql
CREATE FUNCTION
CETE=#

- Se placer dans l'environnement psql, utilisateur postgres : `psql -U postgres CETE`
- Créer toutes les fonctions utiles en base de données :
  - `\i copy_chemins_in_routes.sql`
    Création de la fonction `plpgsql` copy_chemin.
    Copie des enregistrements de la table "chemin_AUtlseD31" vers la table "route_AUtlseD31".
  - `\i insert_mailles.sql`

Création de la fonction `plpgsql` insert_mailles.
Calcul des mailles de la table "route_AUtlseD31" et insertion des mailles dans la table « mailles ».

- `\i insert_pathlink_mission.sql`
  Création de la fonction `plpgsql` insert_pathlink_mission.
  Cette fonction prend en paramètre un id de mission.
  Elle crée les liens inter arrêts physiques pour la mission.
  Remarque : les lignes créées sont uniques. Autrement dit, une seule ligne est créée lorsque deux arrêts sont reliés par plusieurs missions.
- `\i stoparea_calc_stats.sql`
  Création de la fonction `plpgsql` stoparea_calc_stats.
  Cette fonction prend en paramètre une date et une plage horaire..
  Elle calcule les statistiques des arrêts physiques (Cf . IV.A.3.a).
- `\i pathlink_calc_stats.sql`
  Création de la fonction `plpgsql` pathlink_calc_stats.
  Note : la fonction get_average_speed écrite en PL/Python est également insérée en base de données. Le langage PL/Python étant à l'heure actuelle « Untrusted » pour PostGreSQL, cette fonction ne peut être créée que par le superuser postgres.
  Cette fonction prend en paramètre un identifiant de mission, une date et une plage horaire.
  Elle calcule les statistiques des tronçons inter arrêts physiques (Cf.IV.A.3.b).
- `\i insert_mailles.sql`
  Création de la fonction `plpgsql` insert_mailles.
  Calcul des mailles à partir de la table route_AUtlseD31
- `\i insert_deadends.sql`
  Création de la fonction `plpgsql` insert_deadends.
  Calcul des impasses de la table route_AUtlseD31

III. CRÉATION ET INITIALISATION DE LA BASE DE DONNÉES

A. Donneés routières

Dans un premier temps, nous avons travaillé avec les données de la BD Topo fournies par le CETE Méditerranée sont au format natif du logiciel SIG Mapinfo (.tab). Le logiciel SIG libre QuantumGIS est utilisé pour copier les données BD Topo dans la base PostGIS.

Dans une première étape, les données de la BD Topo sont transformées en format shapefile (fichiers .shp). Les fichiers shapefile produits sont ensuite exportés dans la base de données PostGIS.

La couche des routes possède les deux champs « Département_Gestionnaire » et « Département ».

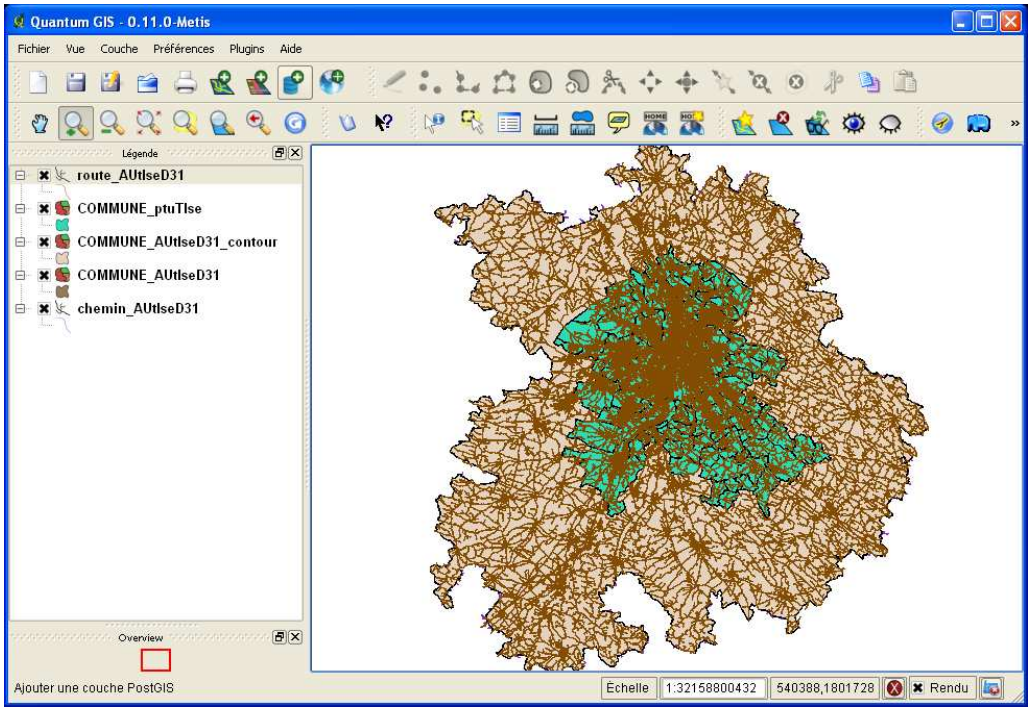
Une fois sauvés au format shapefile, ces champs sont tronqués sur 10 caractères en « Départemen » => deux colonnes possèdent les mêmes noms ce qui provoque une erreur lors de l'import dans la base PostGIS.
Pour contourner ce problème, le shapefile a été généré en utilisant le logiciel ArcGIS en version bureautique d'ESRI qui renomme le deuxième champ Département en « Départem_1 ».

Une procédure similaire permet d'importer la table des routes d'OpenStreetMap à partir de fichiers Shapfile téléchargeables pour la France ou par Région sur un site web OSM.

Le figures ci-dessous représentent les données BD Topo mais le processsus est similaire avec les données OSM.

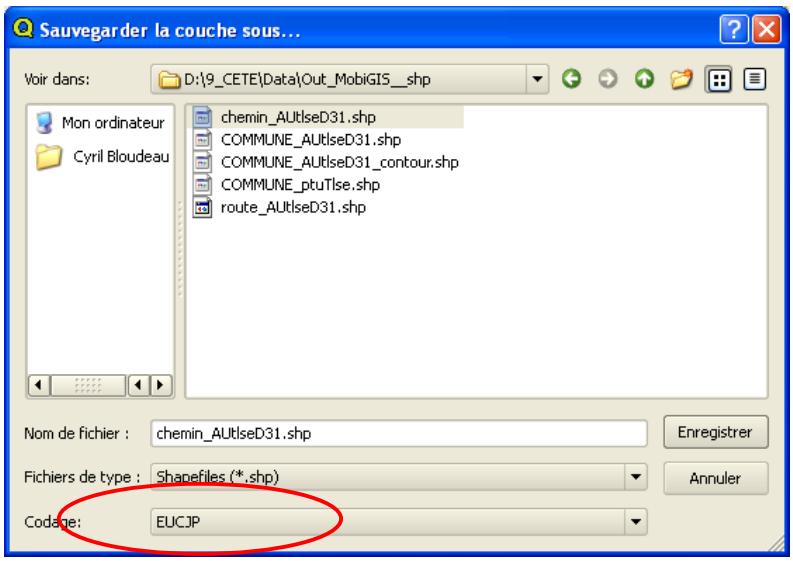
1. Transformation des données en shape files

Dans QuantumGIS, ouvrir les données .tab fournies par le CETE



Clic droit sur une couche et sélectionner « Sauvegarder comme shapefile »

Sauvegarder le fichier en choisissant son nom, son emplacement et son codage

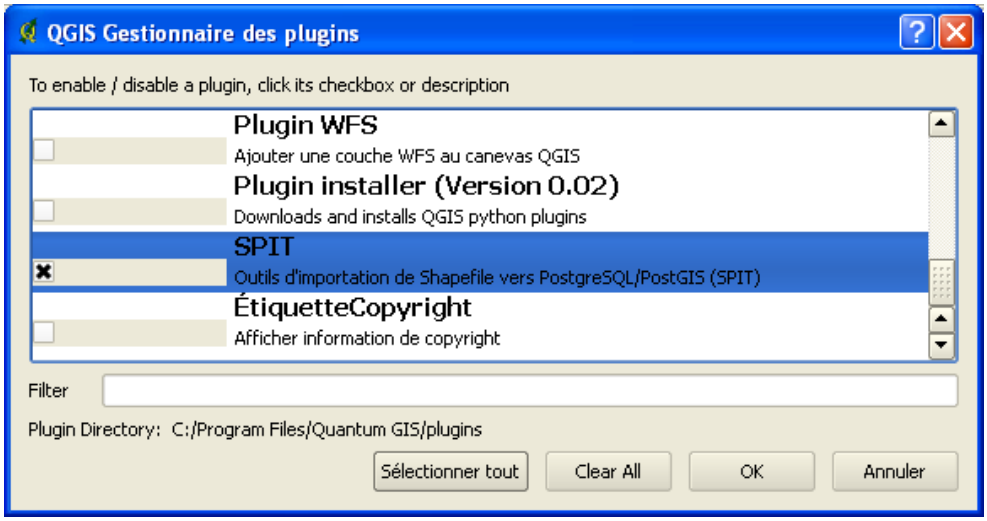


Important : pour permettre une bonne conversion de tous les caractères contenus dans les champs, choisir le **codage : EUCJP**

2. Export des données vers la base PostGIS « CETE »

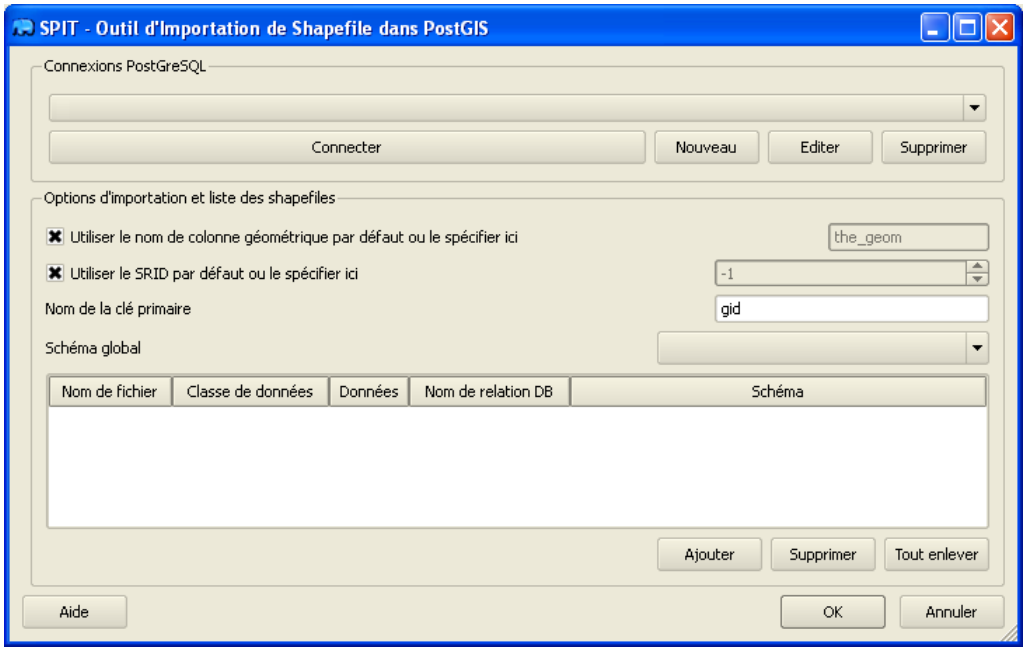
a) Installation du plugin SPIT

Dans la barre de menu de QuantumGIS, cliquer sur Plugins => Gestionnaire de Plugins. Sélectionner l'extension « SPIT » (export vers PostgreSQL/PostGIS)

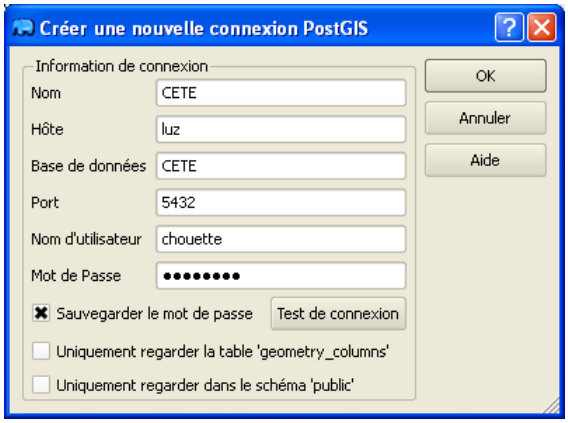


b) Export des données vers PostGIS

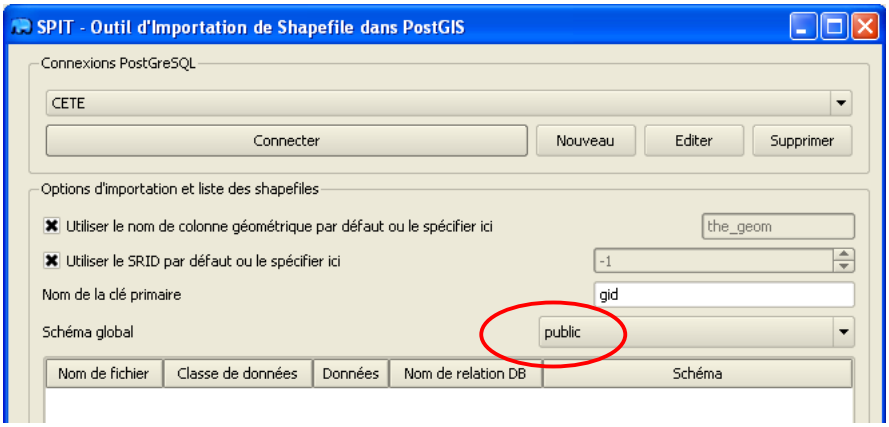
Cliquer sur l'icône pour ouvrir la fenêtre de paramétrage de l'export vers PostgreSQL/PostGIS



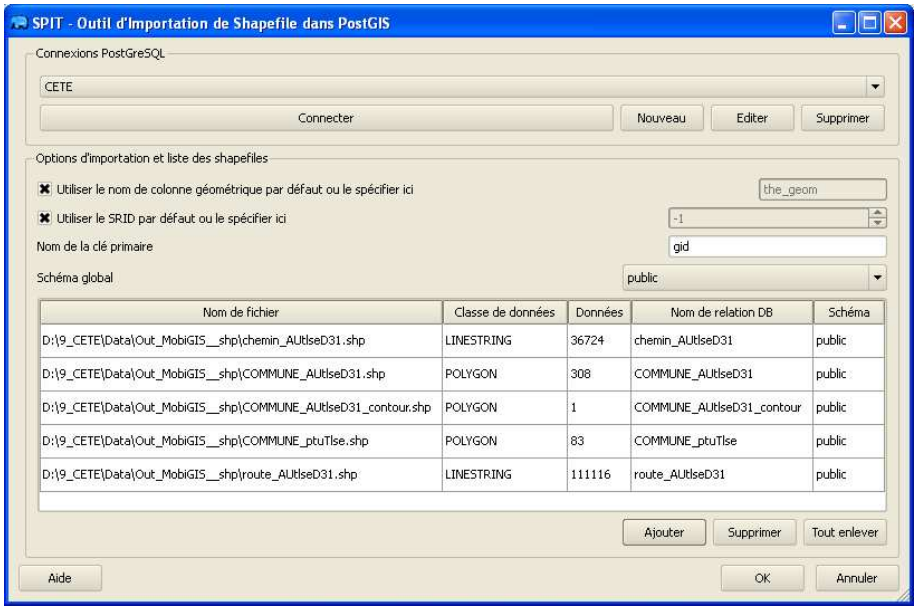
Cliquer sur Nouveau pour ouvrir la fenêtre de paramétrage de la connexion vers la base « CETE », renseigner les différentes informations, tester la connexion et valider



En cliquant sur « Connecter », le schéma global dans lequel vont être ajoutées les données apparaît

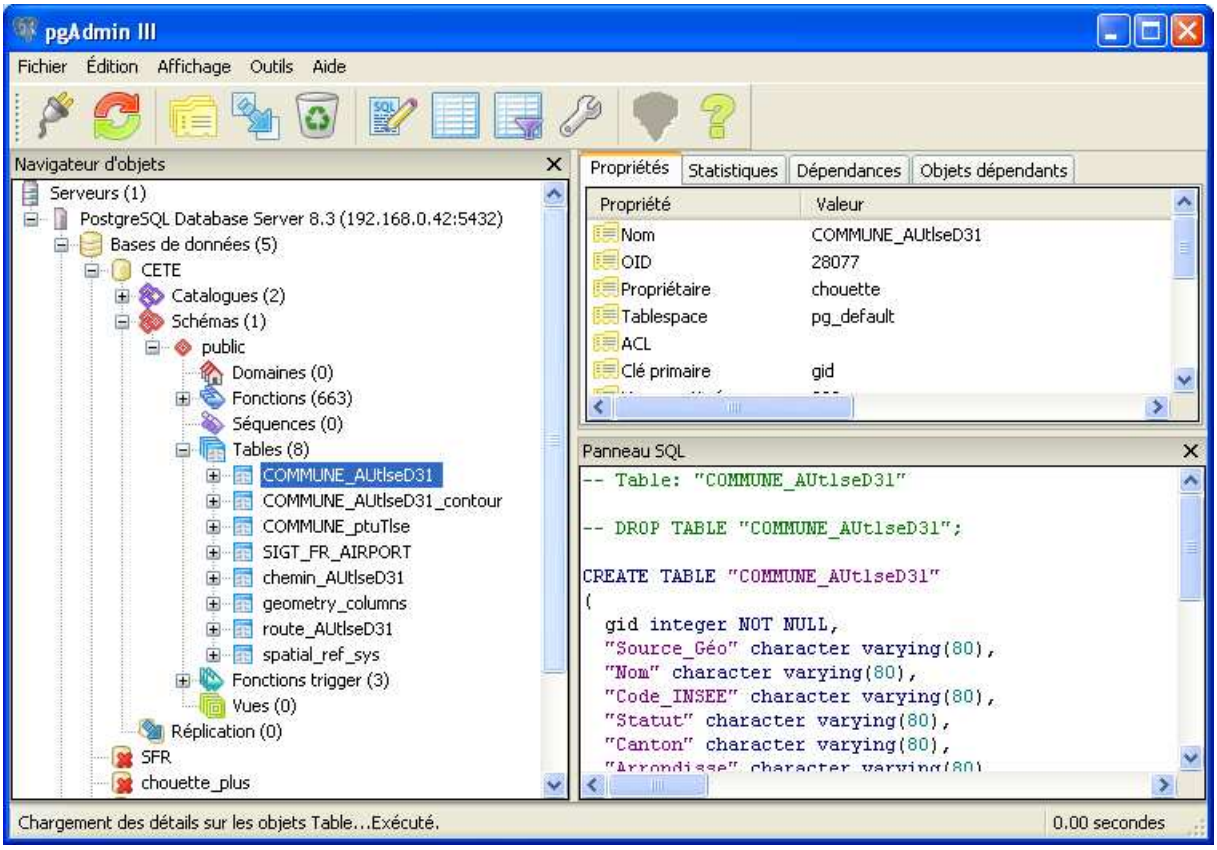


Cliquer sur ajouter, sélectionner les shapefiles à ajouter à la base de données et valider. Les données à charger apparaissent



3. <u>Vérifier l'export des données vers la base « CETE »</u>

Ouvrir PG Admin et sélectionner la base de données « CETE ». Vérifier la présence des tables dans CETE => Schémas => public => tables



4. Copie des chemins dans la table des routes

Dans une fenêtre psql : `select copy-chemin();`

Attention, cette commande ne doit être exécutée qu'une fois, sous peine de voir les chemins dupliqués dans la table des routes.

5. Ajout de la colonne impasse à la table route_AUtlseD31

Dans une fenêtre psql : `ALTER TABLE "route_AUtlseD31" ADD COLUMN deadendflag boolean DEFAULT false;`

Cette commande se trouve également dans le script `create_columns_routes.sql` livré.

B. Données Chouette

Création de la structure de la base de données CHOUETTE.

Les données CHOUETTE peuvent être insérées en base à partir d'un dump PostgreSQL (en utilisant le menu Restaurer de PgAdmin) ou directement à l'aide de l'outil CHOUETTE (cf. documentation du site chouette.mobi).

IV. REQUÊTES DANS LA BASE POSTGRESQL / POSTGIS

A. Requêtes dans la base de données TC

1. Ajout de la structure géométrie aux tables CHOUETTE

Note : la géométrie est intégrée aux versions de CHOUETTE supérieures à la version 1.3 : Les requêtes de création des géométries décrites dans ce chapitre sont inutiles.

De plus, dans ces versions, un trigger met automatiquement à jour la géométrie des éléments lorsque les champs latitude ou longitude sont modifiés dans la base de données PostgreSQL.

a) Table `stoparea`

Le script de création des points est create_geo_stoparea.sql, il contient le code SQL suivant :

```
ALTER TABLE stoparea ADD COLUMN gid serial NOT NULL;
ALTER TABLE stoparea ADD CONSTRAINT unique_gid UNIQUE (gid);

-- Création de la colonne géométrique, EPSG:27572 = Lambert II
SELECT AddGeometryColumn('', 'stoparea','geom',27572,'POINT',2);

CREATE INDEX stopareageom_idx ON stoparea
   USING GIST ( geom GIST_GEOMETRY_OPS );
   -- Création de la colonne journeynb, nombre de courses passant sur une
   période donnée
ALTER TABLE stoparea ADD COLUMN journeynb integer NOT NULL DEFAULT 0 ;
```

Ce script doit être exécuté une et une seule fois, après que la structure de la base de données CHOUETTE ait été crée. Il peut être exécuté dans une fenêtre de commande psql, connecté en tant qu'utilisateur cete :

```
-- \i create_geo_stoparea.sql ;
```

b) Table `pathlink` :

La table pathlink contient des traits rectilignes directs (type « vol d'oiseau ») qui relient deux arrêts physiques (items de la table stoparea) successifs d'au moins une mission.

Deux scripts sont livrés :

- `create_pathlink.sql` : script de création de la table pathlink dans la base de données CHOUETTE

Utilisation :

- Création de la table pathlink.
  Dans une fenêtre psql : `\i create_pathlink.sql ;`

2. Mise à jour des géométries

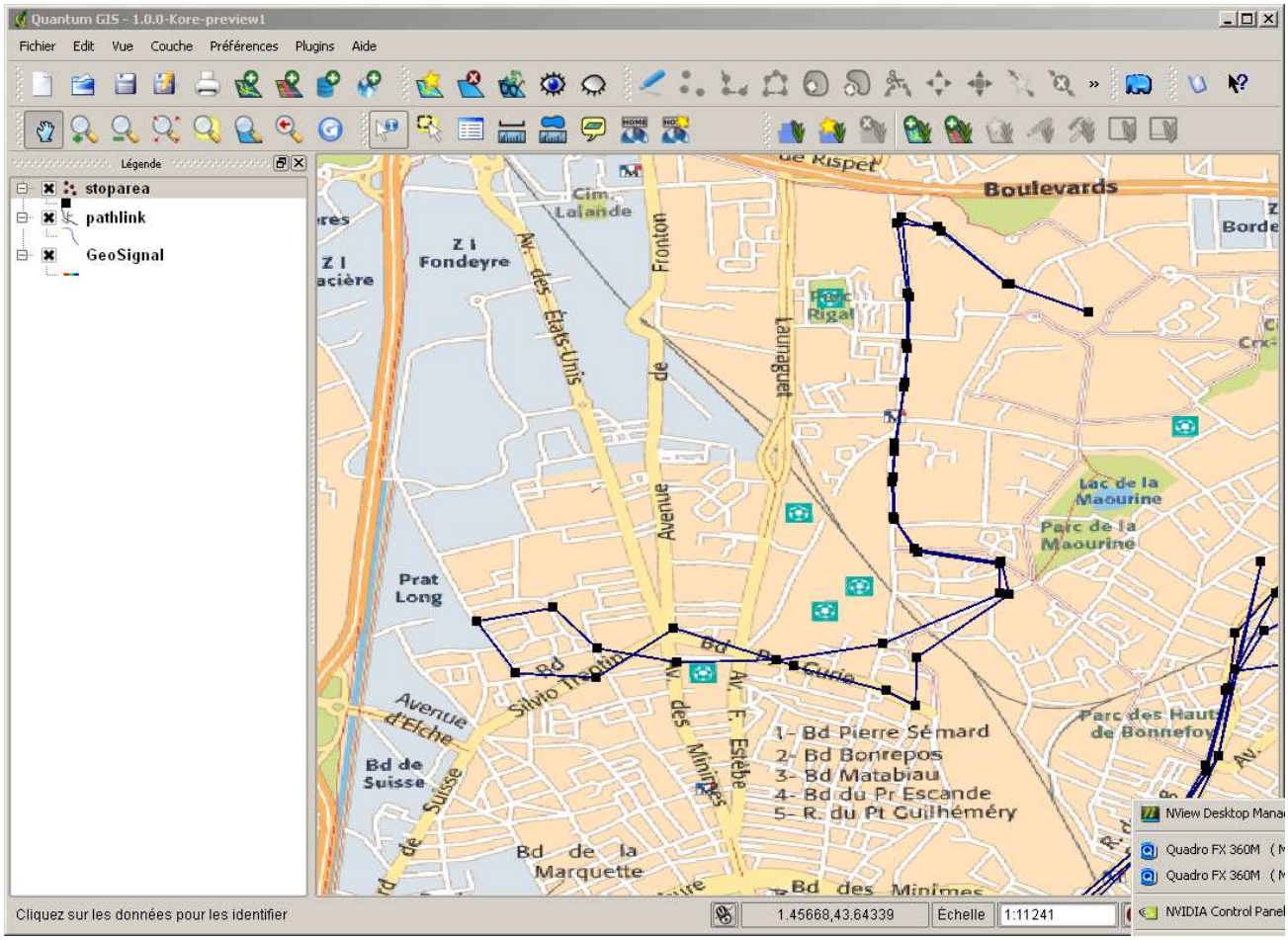
a) Table `stoparea` :

```
-- Maj de la colonne sur la base de logitudes/latitudes
UPDATE stoparea SET geom = PointFromText('POINT(' || x || ' ' || y ||
')',27572) ;
```

b) Table `pathlink` :

```
truncate table pathlink ;
select insert_pathlink_mission(t.id) from journeypattern t order by t.id;
```

Exemple de visualisation dans QGIS :



3. Calcul des indicateurs

Les indicateurs sont calculés sur une période constituée d'une date et une plage horaire : date et horaires sont des paramètres d'entrée de la procédure plsql en charge du calcul des indicateurs.

a) Table `stoparea` :

Un indicateur est calculé pour chaque point d'arrêt :

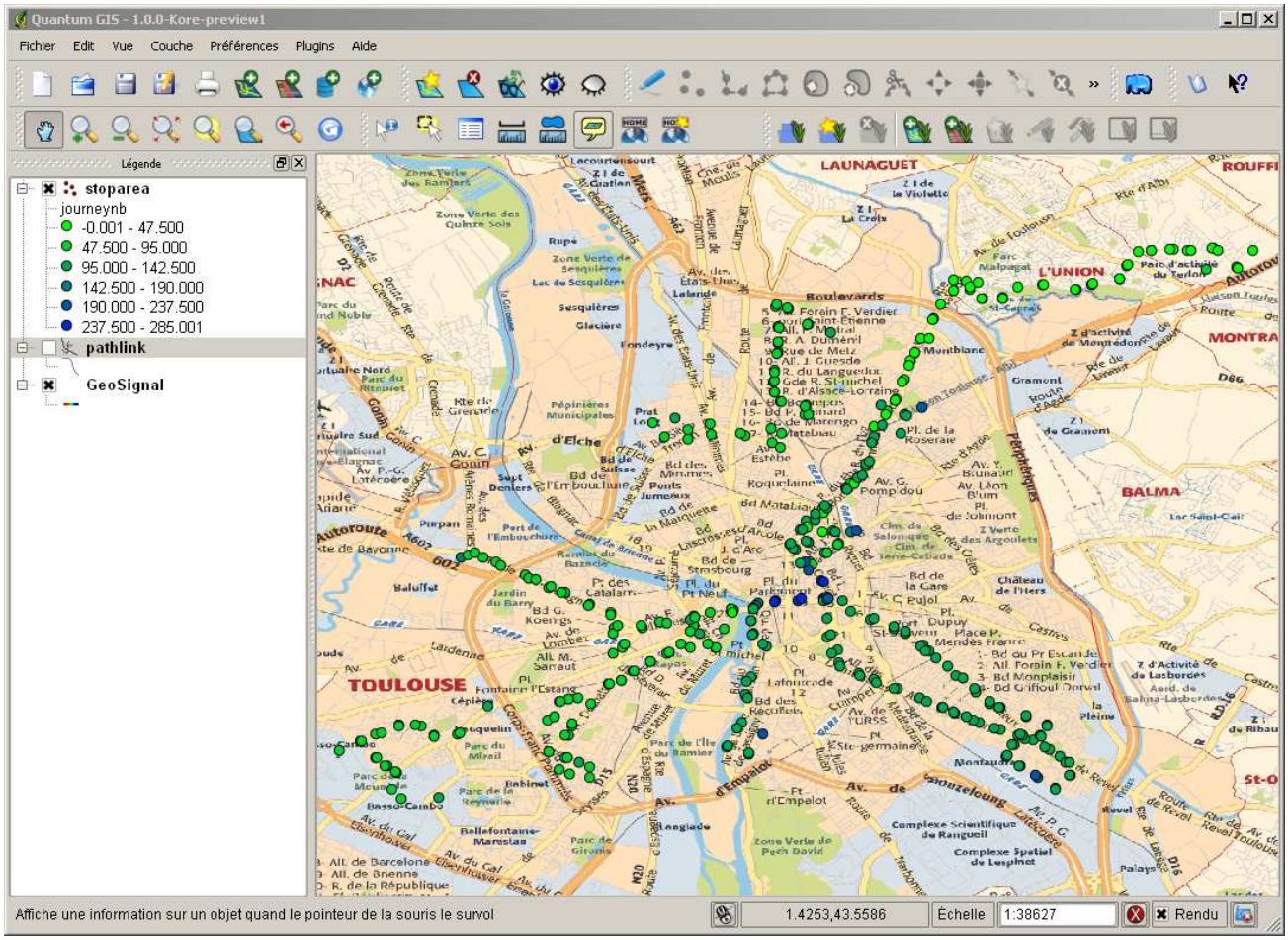
- **journeynb** : nombre de courses passant par le point d'arrêt sur la période

Utilisation :

- Calcul des statistiques pour tous les points d'arrêt :

```
Update stoparea set journeynb = 0 ;
select stoparea_calc_stats('2009-07-27', '05:00:00', '12:00:00') ;
```

Exemple de visualisation dans QGIS pour une requête `select stoparea_calc_stats('2009-27-07', '05:00:00', '12:00:00') ;` :



Remarque : couleurs et plages de valeurs sont définies automatiquement par QGIS pour un nombre de classe de valeurs choisi par l'utilisateur. Il est possible alors, de changer les plages de valeur, de changer les couleurs ou encore, on pourrait choisir un symbole de taille fonction de la fréquence de passage à l'arrêt.
Pour ce dernier point, dans ArcMap d'ArcGIS par exemple et non dans GQIS, un symbole de taille proportionnelle à la fréquence de passage à l'arrêt pourrait être défini.

b) Table `pathlink` :

Les indicateurs sont calculés sur une période constituée d'une date et d'une plage horaire : date et horaires sont des paramètres d'entrée de la procédure plsql en charge du calcul des indicateurs.

Indicateurs calculés pour chaque tronçon:

- **avgspeed** : vitesse moyenne sur le tronçon.

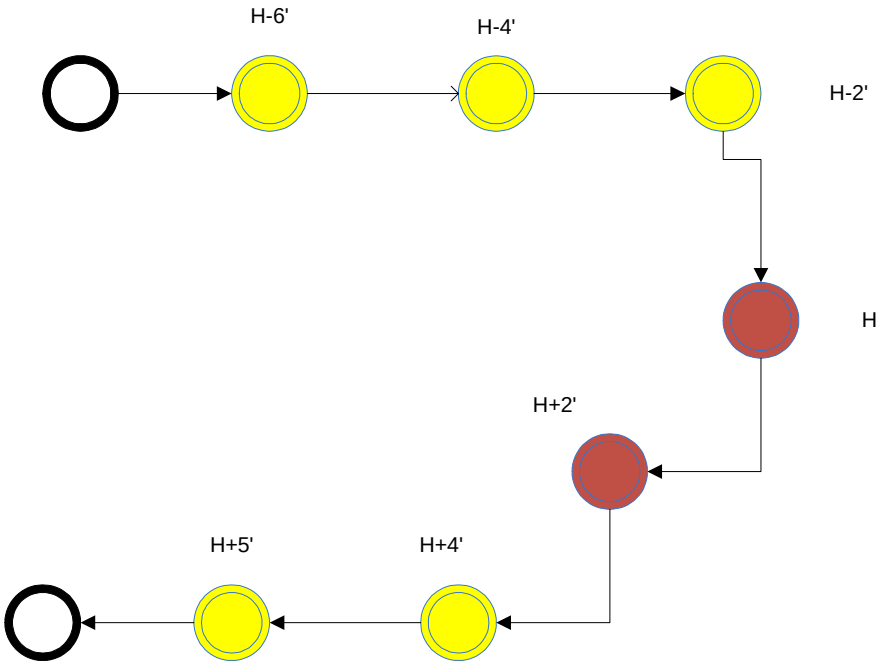
  La moyenne est calculée sur l'ensemble des courses empruntant le tronçon (pathlink) considéré, sur la plage horaire étudiée.

  Dans les faits, les horaires théoriques sont renseignés sur une longueur de trajet significative qui regroupe plusieurs points d'arrêts. Les horaires indiqués entre deux points d'arrêts sont non significatifs (parfois 0 seconde !) ou trop approximatifs (l'unité est la minute). Par conséquent, les horaires ne peuvent être utilisés en l'état pour calculer les vitesses moyennes : il est nécessaire d'élargir le calcul en intégrant les tronçons voisins pour réduire ces approximations.

  La moyenne est calculée sur une « fenêtre de temps » de 10 minutes de temps de trajet autour du tronçon à l'étude ; il est en effet estimé qu'à partir de 10 minutes de temps de trajet, la marge d'erreur liée aux horaires n'influe pas de manière majeure sur les résultats.

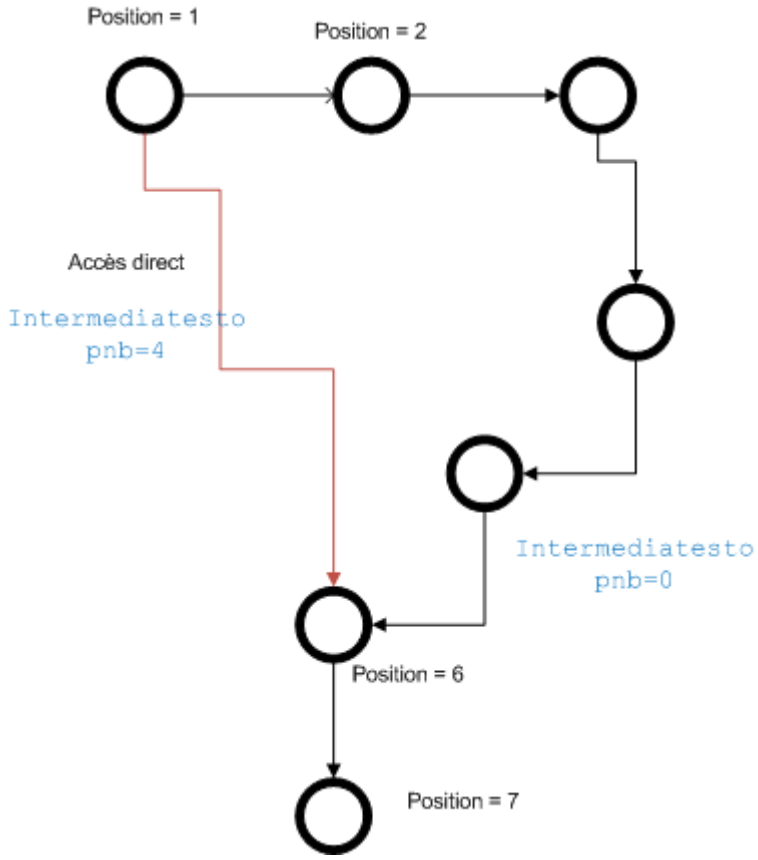
Les moyennes sont calculées à partir de courses qui n'appartiennent pas forcément toutes à la même mission, ... faire le lien avec colonne intermediatestopnb, à quoi sert-elle ?

La vitesse moyenne est calculée sur une zone d'au moins 10 minutes autour du tronçon.
Dans l'exemple ci-dessous, pour calculer la moyenne sur le tronçon H -> H+2 (en rouge) les 7 points d'arrêt colorés en jaune sont pris en compte, la distance est égale à la somme des lignes type vol d'oiseau qui relient les points d'arrêt.



L'ensemble des courses de la période est parcouru : avgspeed est la moyenne des vitesses moyennes obtenues pour chaque course se déroulant dans la période.

- **linknb** : nombre de courses reliant les deux points d'arrêt dans la période.
- **intermediatestopnb** : permet d'identifier les tronçons d'une course de type « direct ».
  En effet, parmi les missions considérées, certaines empruntent des voies directes. Le champ intermédiatestopnb contient le nombre d'arrêts évités par un tronçon direct de ce type de mission comparé à la mission de type « omnibus ».
  Par exemple, Le tronçon ci-dessous où l'on voit 4 arrêts évités par la mission directe se voit attribué un intermediatestopnb de 4.
  Tous les tronçons inter-arrêt de type « Omnibus » ont un intermediatestopnb qui vaut 0.



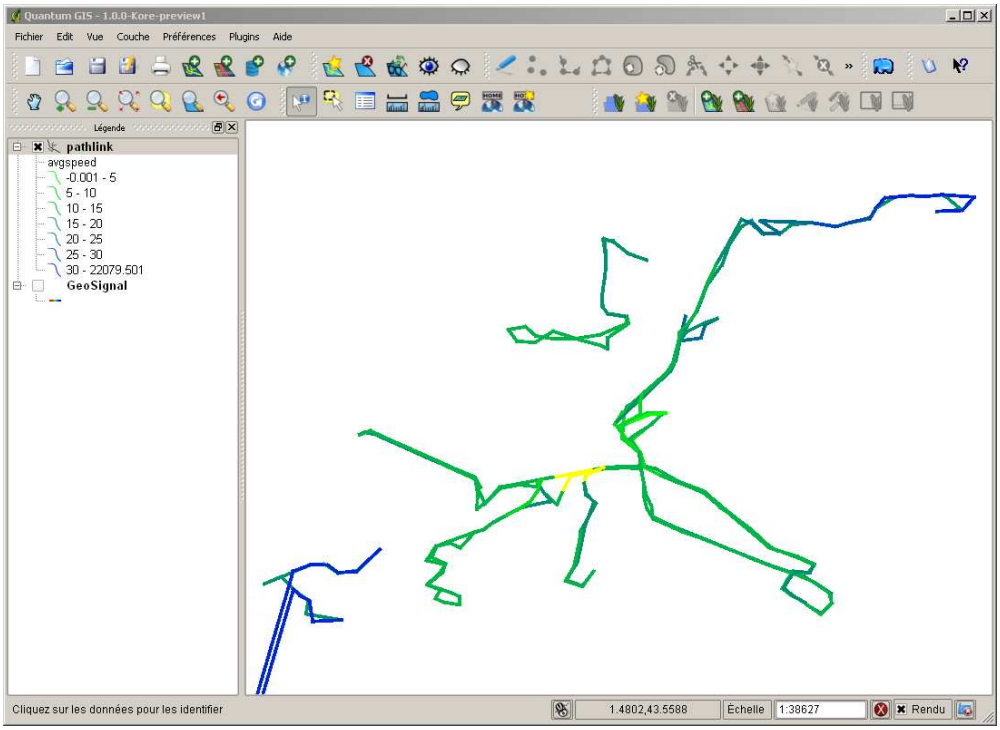
Utilisation :

- Calcul des statistiques pour tous les tronçons :

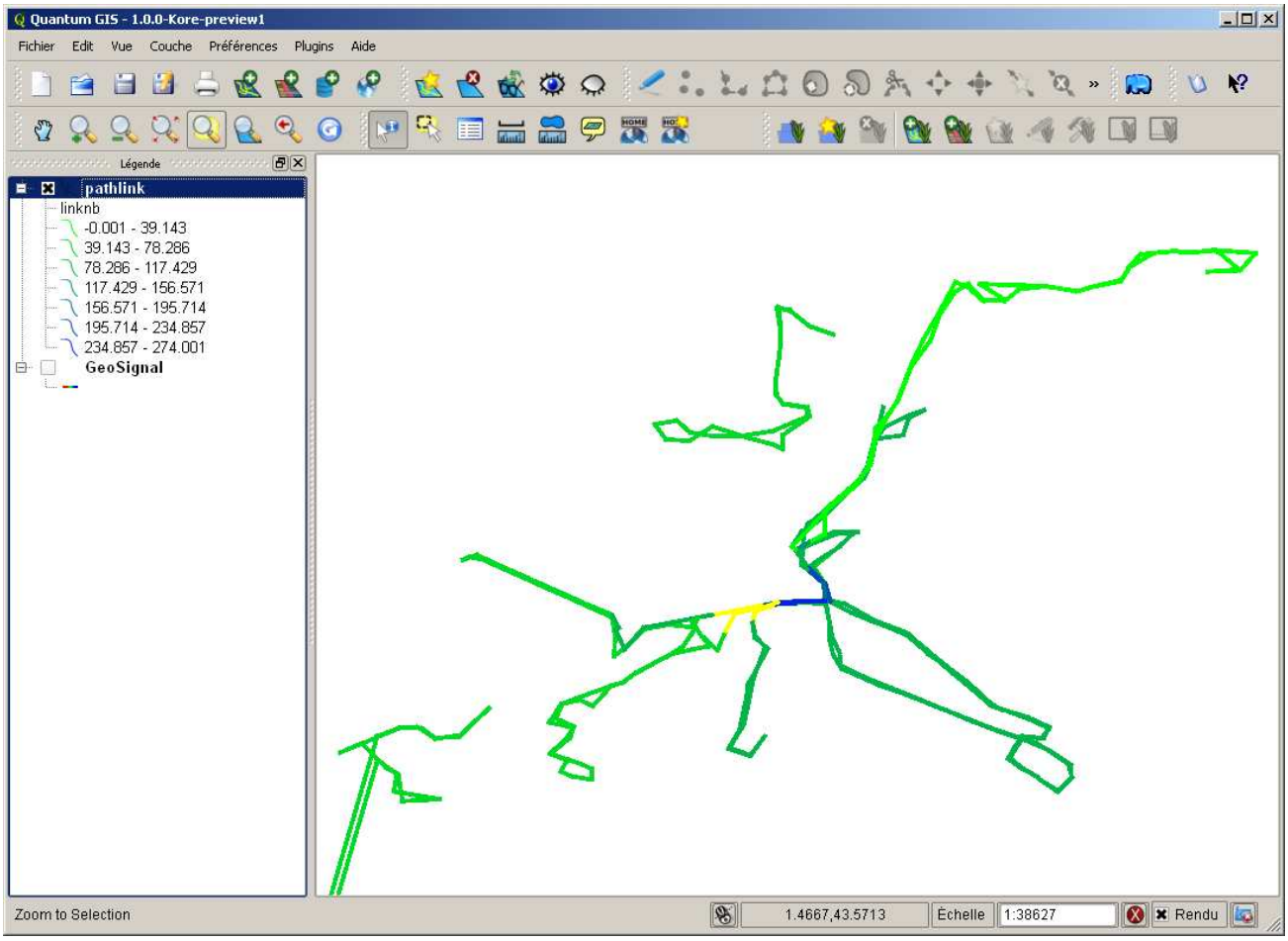
```
Update pathlink set (linknb, intermediatestopnb, avgspeed) = (0, 0, 0) ;

select pathlink_calc_stats(t.id, '2009-07-27', '05:00:00', '12:00:00')
from journeypattern t;
```

Exemple de visualisation dans QGIS pour une requête `select pathlink_calc_stats(t.id, '2009-27-07', '05:00:00', '12:00:00') from journeypattern t;`

Colorisation des tronçons selon la vitesse :



Colorisation des tronçons selon le nombre de courses reliant les deux points d'arrêt :



Remarque : comme pour la colorisation des points d'arrêt les couleurs et plages de valeurs sont définies automatiquement par QGIS pour un nombre de classe de valeurs choisi par l'utilisateur. Il est possible alors, de changer les plages de valeur, de changer les couleurs.
Dans ArcMap d'ArcGIS par exemple, une palette de couleurs peut être choisie pour afficher une graduation complète de couleurs en fonction de la valeur de l'indicateur.

B. Requêtes dans la base de données voirie

1. Copie des données chemin dans la table des routes

Les chemins et les routes sont contenus dans deux tables distinctes extraites de la BD TOPO IGN pour l'aire urbaine de Toulouse, respectivement "chemin_AUtlseD31" et "route_AUtlseD31". Les chemins sont copiés dans la table des routes qui contiendra l'ensemble des données voirie utilisées.

Pour cela, dans une fenêtre psql : `select copy_chemin();`

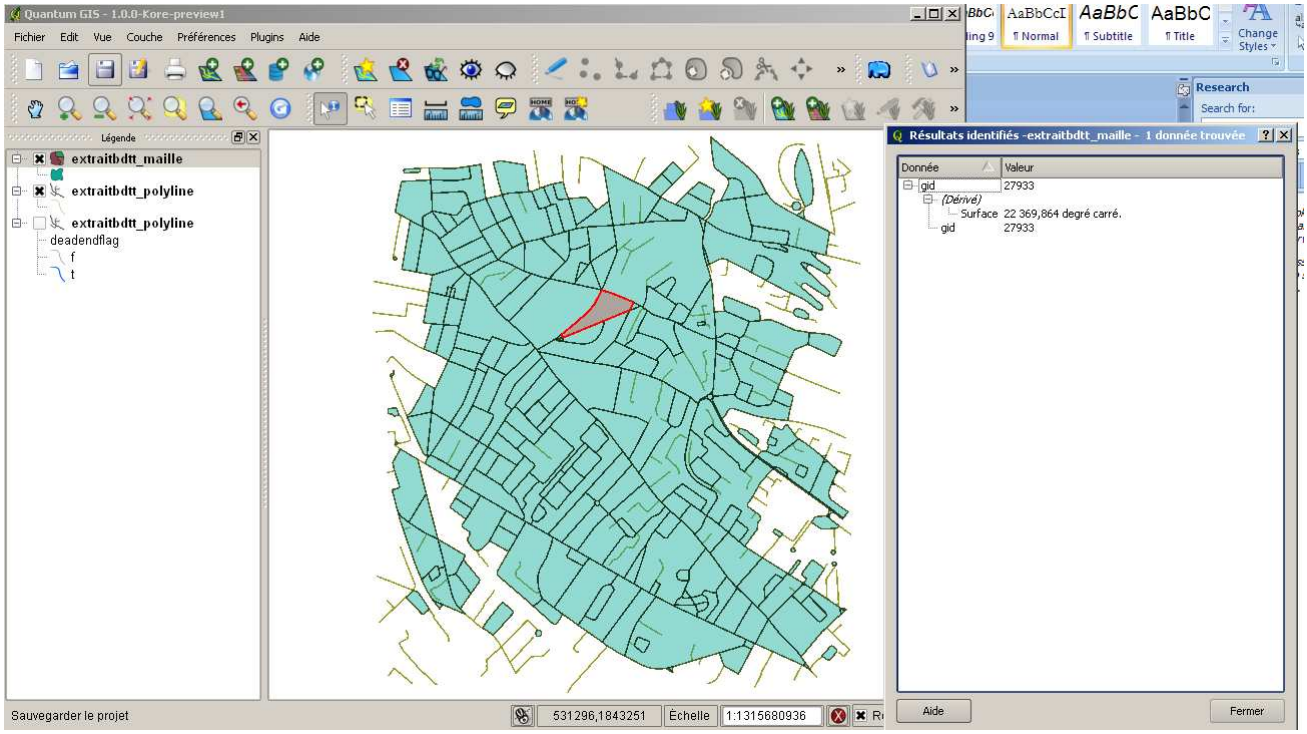
2. Recherche des mailles

Une maille est une liste chaînée d'arcs en boucle, qui part et aboutit au même noeud (un chemin élémentaire dans le vocabulaire de la théorie des graphes).

La recherche des mailles est basée sur la fonction native de PostGIS ST_Polygonize : « *Creates a GeometryCollection containing possible polygons formed from the constituent linework of a set of geometries* ».

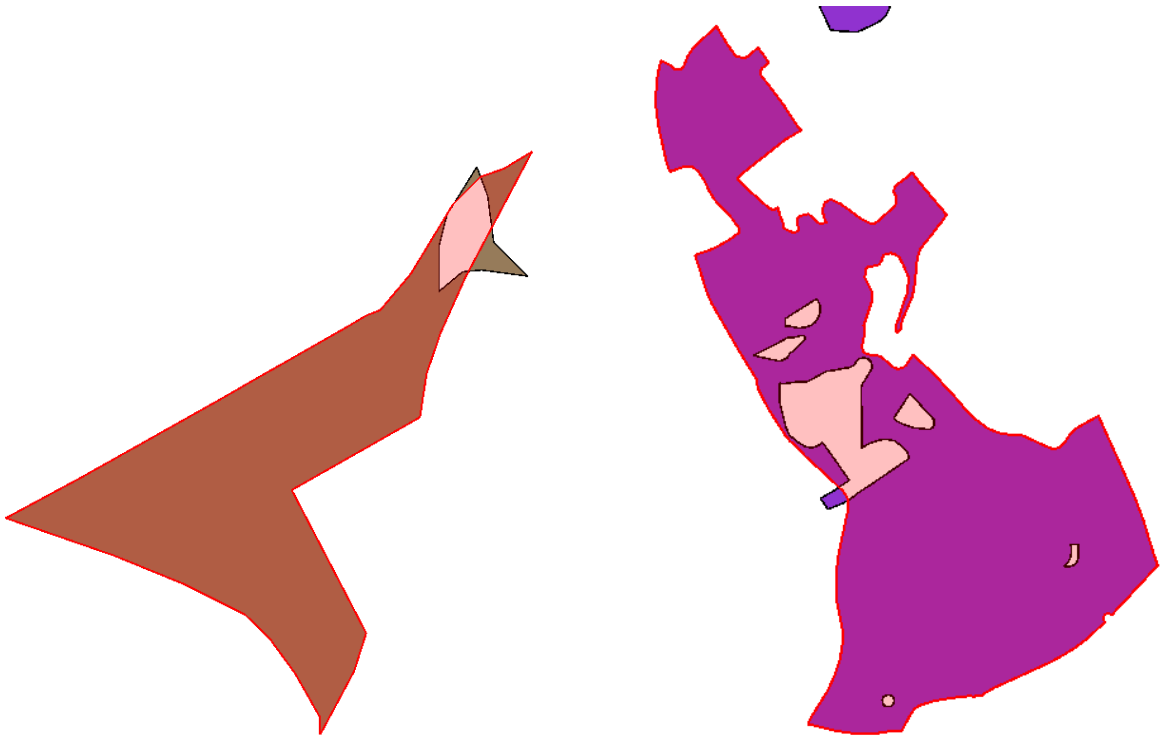
A noter que le calcul des mailles est assez long, environ 1h30 pour la BD Topo de l'aire toulousaine sur un PC XP, Intel Core 2 Duo T7500 2,20 GHz, 2GO RAM.

La copie d'écran ci-dessous présente le résultat de la recherche de mailles pour un extrait de BD Topo. Les polygones constitués apparaissent en noir, les routes ne faisant partie d'aucune maille sont visibles en couleur verdâtre.
La maille en rouge a été sélectionnée.



Pour lancer le calcul des mailles et peupler la table mailles, dans une fenêtre psql : `select insert_mailles();`

A noter, deux polygones calculés par ST_Polygonize ont des géométries non valides :



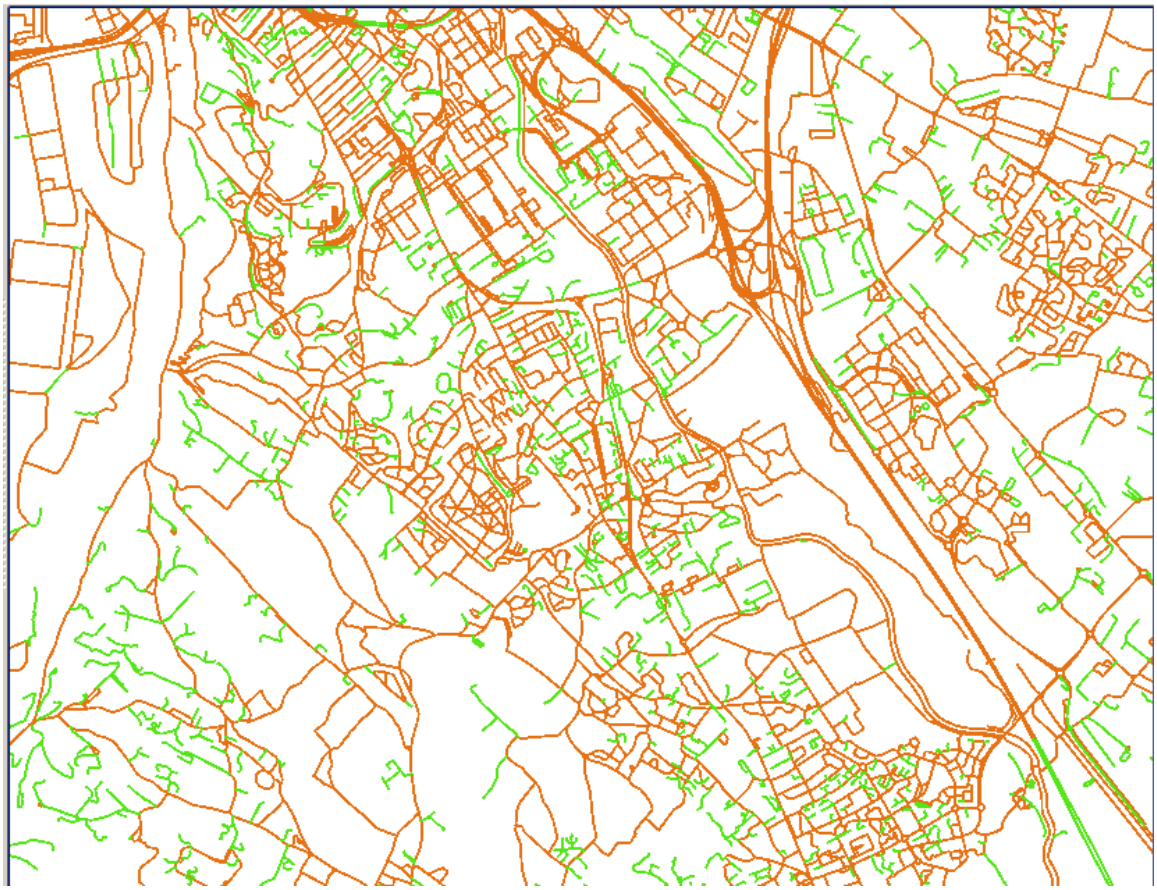
3. Recherche des impasses

Préalable au calcul : les mailles doivent avoir été calculées.

Les arcs entièrement contenus dans les mailles sont recherchés, ils sont marqués comme étant des impasses.

Le calcul des impasses a pris 400 s sur le serveur de données pour l'aire urbaine toulousaine.

La copie d'écran ci-dessous présente le résultat de la recherche d'impasses dans QGIS /



Pour lancer le calcul des impasses et renseigner le champ deadendflag de la table "route_AUtlseD31", dans une fenêtre psql :

```
select insert_deadends();
```

4. Calcul des indicateurs voirie

a) Taux d'impasses

Dans une fenêtre psql :

```
select count(*) from "route_AUtlseD31" where deadendflag=true;
select count(*) from "route_AUtlseD31" where deadendflag=false;
```

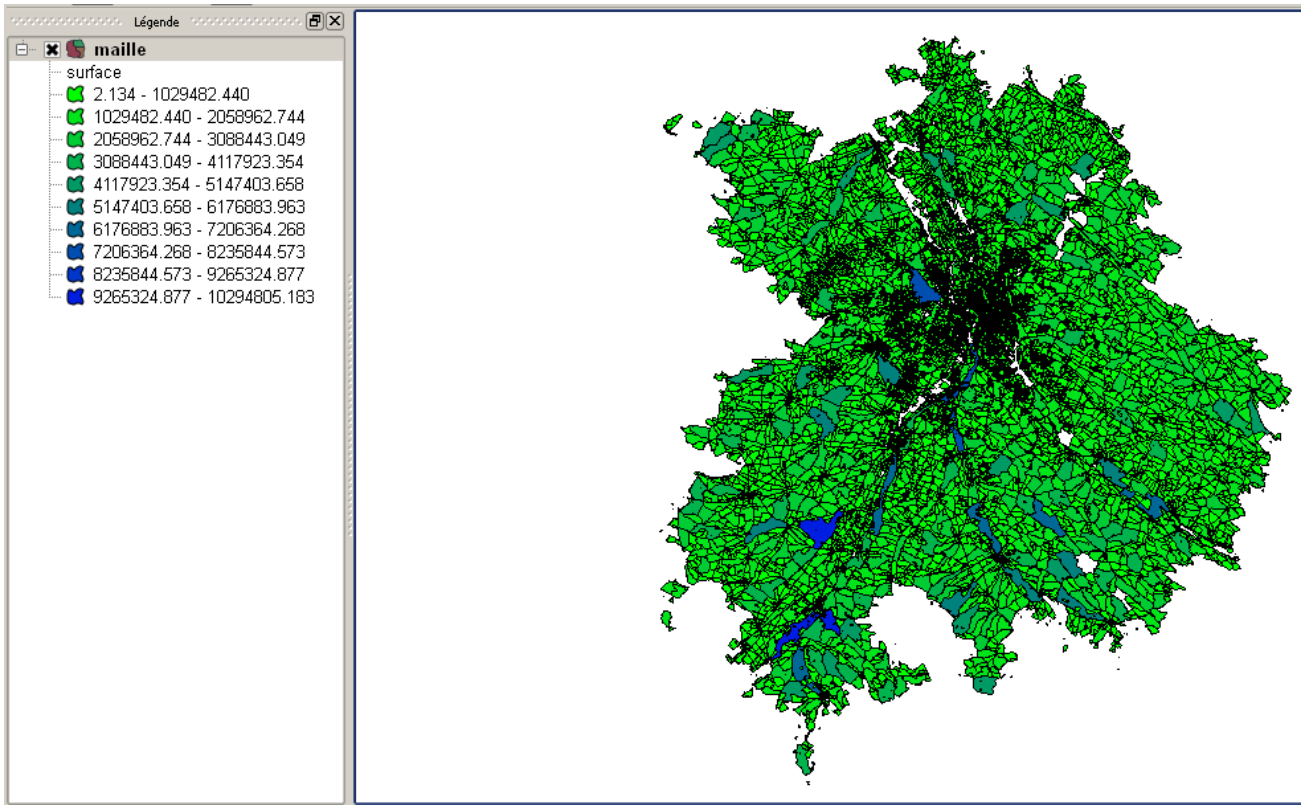
Le nombre d'impasses sur les données de l'agglomération toulousaine est de 37230 **soit 25,1 %.**

b) Distribution de la taille des mailles

La colonne « surface » contient la donnée surface de chaque maille. Le calcul de la surface est réalisé en SQL dans une fenêtre psql :

```
update maille set surface=ST_AREA(geom) ;
```

Un affichage gradué des surfaces peut être réalisé dans GQIS :



Pour calculer les surfaces en ha (et non en m²), il suffit de rajouter « /10000 » dans la requête :

```
update maille set surface=ST_AREA(geom)/10000 ;
```

Pour obtenir un fichier de résultats interprétable dans Excel. Dans une fenêtre psql :

```
\o results.txt
```
-- Les résultats de la commande sont écrits dans le fichier results.txt du répertoire courant

```
select surface, gid from maille;
```

Les données sont contenues dans le fichier results.txt, elles peuvent être interprétées par un outil du type Excel par exemple :

- Ouvrir le fichier results.txt dans Excel en utilisant l'option Delimited, choisir le caractère | comme délimiteur

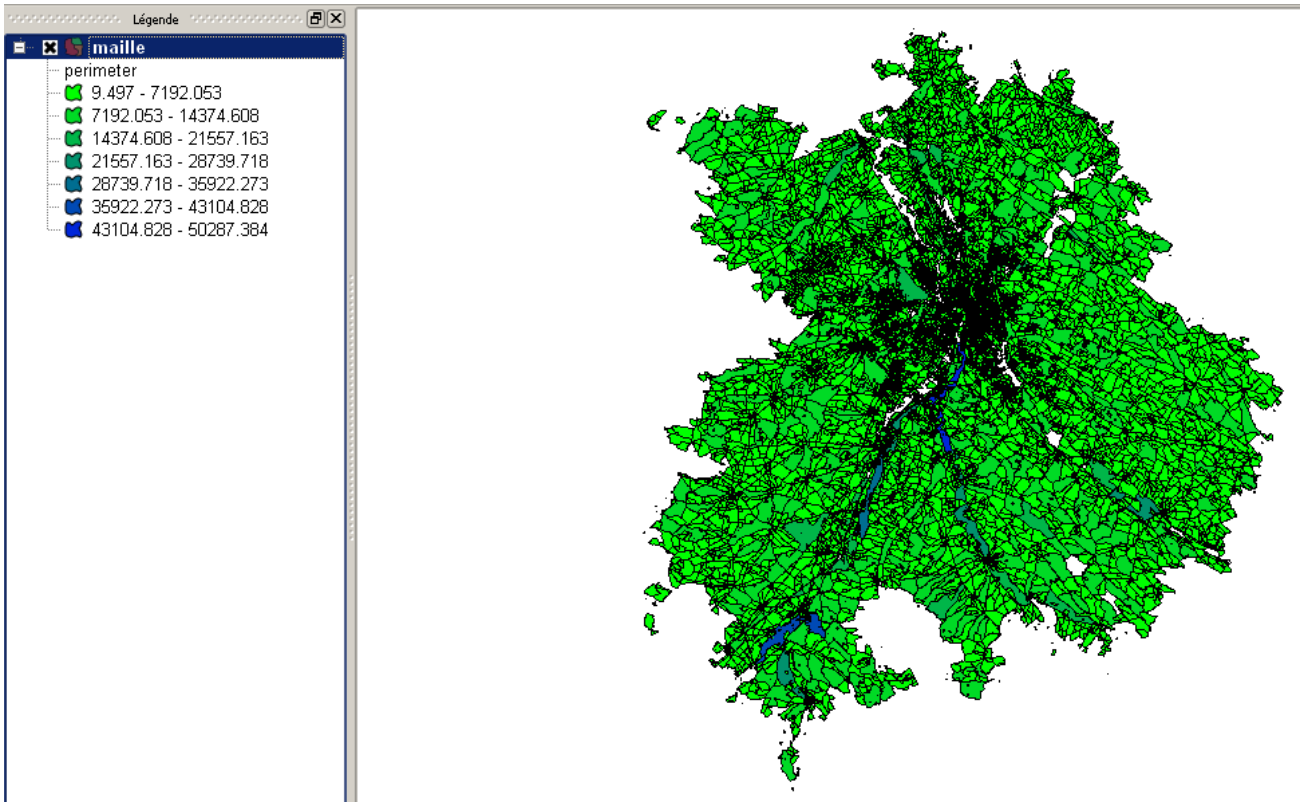
- Créer des graphes sont fournis ci-après :
  - Nombre d'éléments par plage de valeurs (classes de superficie ou de périmètre)
  - Distribution des valeurs (histogramme des surfaces ou des périmètres)
  -

c) Valeurs Min, Max, moyenne et médiane du périmètre des mailles

La colonne « perimeter » contient la donnée périmètre de chaque maille. Le calcul de périmètre est réalisé en SQL dans une fenêtre psql :

```
update maille set perimeter=ST_PERIMETER(geom) ;
```

Un affichage gradué des périmètres peut être réalisé dans GQIS :



Pour calculer les périmètres en km (et non en mètre), il suffit de rajouter « /1000 » dans la requête :

```
update maille set perimeter=ST_PERIMETER(geom)/1000 ;
```

Calcul des indicateurs :

- Valeur Min : `select min(perimeter) from maille;`
- Valeur Max : `select max(perimeter) from maille;`
- Valeur médiane (attention, le calcul est un peu long, il a pris 574 s sur un serveur Intel XEON 4 cœurs) :

```
CREATE OR REPLACE FUNCTION occurrences_cumulees(FLOAT) RETURNS INTEGER
AS
      'SELECT sum(1)::INTEGER FROM maille where perimeter <= $1;'
 LANGUAGE SQL;

CREATE OR REPLACE FUNCTION Moitie_population() RETURNS INTEGER AS
      'SELECT count(*)::INTEGER/2 FROM maille;'
 LANGUAGE SQL;

SELECT min(perimeter) AS median FROM maille
   WHERE occurrences_cumulees(perimeter) >= Moitie_population();
```

- Valeur moyenne : `select avg(perimeter) from maille;`

V. DOCUMENT APPLICABLE ET RÉFÉRENCE

[DA1] Cahier des charges Requêtes BD SIG SIG IMM, version 0.4 du 10/09/08 (Consultation en vue d'une Prestation Intellectuelle du CETE Méditerranée)

MOBIGIS, Requetes SIG VP/TC, Rapport Phase 1 pour CETE Med., V3 du 27/07/09.

Conception d'un outil open source d'analyse SIG voirie/TC, Claire Christol, Rapport de stage Mastère SIG Université de Provence / Digne, CETE Méditerranée, Juin 2009, http://www.cete-mediterranee.fr/tt13/www/article.php3?id_article=192

Site Internet :

www.potimart.org

www.predim.org

www.cete-mediterranee.fr/tt13

VI. GLOSSAIRE

BD	Base de Données
IMM	Information Multi Modale
QGIS	Quantum GIS Open Source
PREDIM	Plate-forme de Recherche et d'Expérimentation pour le Développement de l'Information Multimodale
SIG	Système d'Information Géographique